RICE UNIVERSITY

Cooperative Partial Detection for MIMO Relay Networks

by

Kiarash Amiri

A Thesis Submitted

in Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

Approved, Thesis Committee:

Joseph R. Cavallaro, Chair
Professor of Electrical and Computer Engineering and Computer Science

---

Behnaam Aazhang
J.S. Abercrombie Professor of Electrical and Computer Engineering

---

Ashutosh Sabharwal
Associate Professor of Electrical and Computer Engineering

Mark Embree
Professor of Computational and Applied Mathematics

Jorma Lilleberg
Adjunct Professor of Electrical and Computer Engineering

Houston, Texas

November, 2010

ABSTRACT

Cooperative Partial Detection for MIMO Relay Networks

by

Kiarash Amiri

Cooperative communication has recently re-emerged as a possible paradigm shift to realize the promises of the ever increasing wireless communication market; however, there have been few, if any, studies to translate theoretical results into feasible schemes with their particular practical challenges. The multiple-input multiple-output (MIMO) technique is another method that has been recently employed in different standards and protocols, often as an optional scenario, to further improve the reliability and data rate of different wireless communication applications. In this work, we look into possible methods and algorithms for combining these two techniques to take advantage of the benefits of both.

In this thesis, we will consider methods that consider the limitations of practical solutions, which, to the best of our knowledge, are the first time to be considered in this context. We will present complexity reduction techniques for MIMO systems in cooperative systems. Furthermore, we will present architectures for flexible and

configurable MIMO detectors. These architectures could support a range of data rates, modulation orders and numbers of antennas, and therefore, are crucial in the different nodes of cooperative systems. The breadth-first search employed in our realization presents a large opportunity to exploit the parallelism of the FPGA in order to achieve high data rates. Algorithmic modifications to address potential sequential bottlenecks in the traditional bread-first search-based SD are highlighted in the thesis.

We will present a novel Cooperative Partial Detection (CPD) approach in MIMO relay channels, where instead of applying the conventional full detection in the relay, the relay performs a partial detection and forwards the detected parts of the message to the destination. We will demonstrate how this approach leads to controlling the complexity in the relay and helping it choose how much it is willing to cooperate based on its available resources. We will discuss the complexity implications of this method, and more importantly, present hardware verification and over-the-air experimentation of CPD using the Wireless Open-access Research Platform (WARP)

Acknowledgments

I would like to start by thanking my advisor and mentor, Prof. Joseph R. Cavallaro for his thoughtful comments, support and guidance during these five years of graduate study at Rice University. Prof. Cavallaro has shown great care and concern for his graduate students, and have always helped them advance and become not only better researchers, but more importantly, successful individuals in their individual and professional lives. He was my first mentor and advisor when I entered this country, and I would not have been able to successfully finish my PhD if it were not for his detailed guidance.

I would also like to express my sincere appreciation to the department chair, Prof. Behnaam Aazhang, for leading this department during my graduate study at Rice. I am thankful for Prof Aazhang, Prof. Cavallaro and Prof. Sabharwal efforts in creating and maintaining a unique environment in the CMC lab where I have had the chance to think, learn and experiment the work some of which is presented in this thesis. My committee, Prof. Joseph R. Cavallaro, Prof. Behnaam Aazhang, Prof. Ashutosh Sabharwal, Prof. Mark Embree and Dr. Jorma Lilleberg have provided great guidance to help me complete this thesis and research problem. Dr. Chris Dick has always been a great support, and I am glad I have had the opportunity to work in his group, and collaborate with him subsequently. This research thesis would

not have been possible without the generous support from NSF grants EIA-0321266, CCF-0541363, CNS-0551692, CNS-0619767, EECS-0925942, and CNS-0923479, as well as the support from Nokia, Xilinx, Nokia Siemens Networks, Texas Instruments, and Azimuth Systems.

My sincere thanks go to my parents, Zahra and Mansour, for being helpful throughout all these years, and their support in all the aspects of my life. They dedicated their lives to the education and advancement of their children, and I am, and will forever be, grateful for that. I am also grateful to my brother, Keyvan, for always challenging me to be more productive and accurate. I would not have envisioned achieving my academic goals without my family's aid.

I should also thank my close friend, Farbod, who has had deep impact on my life and my research approach. I am also thankful for the friends that I have had the chance of interacting and collaborating with over the past five years at Rice. I can not name all of them; however, I have to mention a few (in no particular order): AhmadR, Borghan, Marjan, Mehrdad, Ehsan, Bei, Guohui, Yang, MikeW, MikeB, ChrisS, Predrag, Gareth, Debashish, Arnab, Jaska, DavidK, Melissa, Nahal, AmirP, Patrick, Sid, MonaSh, Layla, SamS, Manjari, ...

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

MIMO Systems: Multiple-input multiple-output (MIMO) communication systems and spatial division multiplexing (SDM) have recently drawn significant attention as means to achieve tremendous gains in system capacity and link reliability. Moreover, spatial division multiple access (SDMA) has recently received attention for its promise to increase the sum data rate of different users in wireless networks, and creating a virtual MIMO between multiple users and a base station.

The optimal hard decision detection, in terms of Bit Error Rate (BER) performance, for all MIMO wireless systems is the maximum likelihood (ML) detector. However, the cost of direct implementation of ML grows exponentially with the number of antennas and the modulation scheme, making its ASIC or FPGA implementation infeasible for all but low-density modulation schemes using a small number of antennas. Sphere detection (SD) solves the ML detection problem in a computationally efficient manner [107, 34, 8, 60, 59].

A flexible architecture detector that can support various schemes needs to be an integral part of any MIMO relay network. Depending on the signaling structure, the relay may need to switch between different modulation orders. Therefore, the architecture of the MIMO detector in the relay needs to support different

number of antennas and modulation orders, and be able to change the configuration accordingly on-the-fly.

Cooperative Systems: While the dedicated multi-antenna relays will be capable of performing computationally intensive operations, potential MIMO relays will be mobile multi-antenna users that could choose to assist the active links in the environments during their idle time. One major criteria for such idle MIMO users to act as relays is to ensure that such cooperation will not require significant processing battery power that they would need later for their own use. Full detect-and-forward in the relay can require a significant amount of resources in MIMO cooperative communications, particularly if the relay chooses to perform a close-to-optimum detection. This effect becomes more important when one considers the practical resource constraints of idle MIMO users operating as relays. Therefore, it is crucial to distribute the detection task between the relay and the destination in such a way that the relay does not need to spend too much of its processing and transmit power, and yet, can enhance the performance compared to a non-relay scenario.

Another challenge in studying relay networks and cooperative communication, particularly in the case of multiple multi-antenna nodes in a MIMO relay scenario, is the impractical assumption of full channel state information in the source, relay and destination.

More often than not, the wireless channel is not reciprocal; thus, the transmitter

does not have complete knowledge of the channel unless a high rate feedback link transmits the channel coefficients back to the transmitter. Using feedback to transmit the complete channel knowledge can be a daunting and costly overhead in MIMO systems. Moreover, deployment of relays further increase the interference level in the wireless networks. The extra interference in the network environment is particularly challenging because the current wireless systems suffer from large interference due to other users and cells.

1.1.1 Scope of the Thesis

The problem of sharing the resources between the nodes in a cooperative way has several dimensions. The main factors in understanding this space are the complexity of operations and the performance/rate gains obtained through resource sharing and cooperation. For instance, a high complexity scheme could consist of several nodes fully dedicated to detection and decoding of the source message(s). A lower complexity scheme can consist of one (or multiple) nodes simply forwarding the received signal without performing any signal processing operation on the node. Moreover, a relay node could decide to jointly, or separately, code its messages with the source and transmit the joint message to the destination. Therefore, the source(s) and relay(s) have a wide range of options for cooperating and transmitting their messages to the destination, creating a large design space.

Due to the very large space and set of options, finding a universal solution could

be extremely difficult. Therefore, in this thesis, we focus our attention on answering specific scenarios that are more common, and more importantly, provide insights for answering the other cases; hence, helping the overall effort of understanding the larger design space. We will consider scenarios that include one source and one relay, and, we specifically focus on schemes that are similar to decode-and-forward since such schemes pose the most challenge for maintaining complexity in the relay. Also, we will look at multi-antenna nodes, since performing multi-antenna detection adds one more level of freedom in the relay and destination, while adding up to the complexity in the relay.

While looking at these cases will help cover an important part of the overall complexity-performance space, there are other cases that this thesis will not focus on. In particular, we will not consider multi-source, multi-relay settings. We do not consider cases where the relay combines its own message with the source's message, and we primarily rely on channel coding schemes that are currently used in the wireless standards to support backward compatibility. However, while our solution does not address these cases, it will provide insights that will help addressing those cases as well.

System Model Definitions

Throughout this thesis, we assume a three node network: the source, relay and destination, denoted by S, R and D; respectively. We further assume that the source,

relay and destination are equipped with M_s, M_r and M_d antennas; respectively. Given the practical limitations of deploying full duplex radios, we assume the relay operates in half-duplex mode. The communication between the source and the destination is performed over two *time slots*. In the first time slot, the source broadcasts its message to both the relay and the destination; and in the second time slot, the relay, using M_r of its antennas, transmits its message to the destination while the source is silent.

We assume coded systems, where the bits are coded and spread across the transmit antennas in the source before modulation. The bits b_j, $j = 1, ..., K$ are passed through the channel coder of rate K/N in the source node to generate u_l, $l = 1, ..., N$. The u_l bits are mapped to modulation points x_i and spread across the transmit antennas of the source to form the source transmit vector $\mathbf{x}_s = [x_1, x_2, ..., x_{M_s}]^T$.

The received signals at the relay and destination at the end of the first time slot are given by

$$\mathbf{y}_r = \mathbf{H}_{sr}\mathbf{x}_s + \mathbf{n}_r \tag{1.1}$$

$$\mathbf{y}_d^{(1)} = \mathbf{H}_{sd}\mathbf{x}_s + \mathbf{n}_d^{(1)}. \tag{1.2}$$

The relay, then, detects all or part of the transmitted vector symbols, and forwards them to the destination. Therefore, the received signal at the destination at the end of the second time slot is given by

$$\mathbf{y}_d^{(2)} = \mathbf{H}_{rd}\mathbf{x}_r + \mathbf{n}_d^{(2)}, \tag{1.3}$$

where superscripts $^{(1)}$ and $^{(2)}$ are used to distinguish the first and second time slots. Since the relay receives only at the end of the first time slot, no superscript is used for the relay. The noise vectors, $\mathbf{n}_r$, $\mathbf{n}_d^{(1)}$ and $\mathbf{n}_d^{(2)}$ are of size M_r, M_d and M_d, with each of their elements chosen from a complex symmetric Gaussian variable $\mathcal{C}N(0,1)$. We also assume that each element of the $\mathbf{x}_s$, $\mathbf{x}_r$ and $\mathbf{x}_d$ vectors are chosen from a QAM modulation, $\mathcal{O}$, with the modulation set size of $w = |\mathcal{O}|$, and average power constraint of $E[{x_i}^2] = 1$.

Note that the type of processing in the relay depends on the amount of available resources in the relay. The relay can choose the detection process, and how much it is willing to detect the transmitted signals, and whether or not, it should perform decoding and re-encoding of the transmitted signals. This is one of the contributions of this chapter, and will be discussed in more detail in the next sections.

As illustrated in Figure 1.1, the $\mathbf{H}_{sr}$, $\mathbf{H}_{rd}$ and $\mathbf{H}_{sd}$ are matrices of sizes $M_r \times M_s$, $M_d \times M_r$ and $M_d \times M_s$, and correspond to the channel matrices between the source and the relay, relay and the destination, and source and the destination, respectively. All these channel matrices, $\mathbf{H}_{sr}$, $\mathbf{H}_{rd}$ and $\mathbf{H}_{sd}$, have independent elements, each drawn from a circularly symmetric Gaussian random distribution with zero mean and vari-

ances of σ_{sr}^2, σ_{rd}^2 and σ_{sd}^2, respectively, where:

$$\sigma_{sr}^2 = \sqrt{\frac{\text{SNR}_{sr}}{M_s}} \tag{1.4}$$

$$\sigma_{rd}^2 = \sqrt{\frac{\text{SNR}_{rd}}{M_r}} \tag{1.5}$$

$$\sigma_{sd}^2 = \sqrt{\frac{\text{SNR}_{sd}}{M_s}}. \tag{1.6}$$

We make the practically feasible assumption that the $\mathbf{H}_{sr}$ matrix is known in the relay, and the $\mathbf{H}_{sd}$ and $\mathbf{H}_{rd}$ matrices are known in the destination node; thus, only the receivers of each communication link have complete channel knowledge.

The signal-to-noise ratios, SNR, at each of the received antennas of the relay and destination are defined as

$$\text{SNR}_{sr} = \frac{\mu P}{(d_{sr})^{\alpha}}, \tag{1.7}$$

$$\text{SNR}_{rd} = \frac{(1-\mu)P}{(d_{rd})^{\alpha}} \tag{1.8}$$

$$\text{SNR}_{sd} = \frac{\mu P}{(d_{sd})^{\alpha}}, \tag{1.9}$$

where α is the path loss exponent, which usually varies between 2 and 6. The above SNR equations imply that the sum transmit power from the source and the relay is set to P, and is split with a proportionality factor of $0 < \mu \leq 1$, such that the source uses μP and the relay uses $(1-\mu)P$. Therefore, if τ represents the symbol time, then the amount of energy per information bit is given by

$$E_b = \frac{\tau\mu P + \tau(1-\mu)P}{M_s \log w(K/N)} = \frac{\tau P}{M_s \log w(K/N)} \text{ [Joules/bit]} \tag{1.10}$$

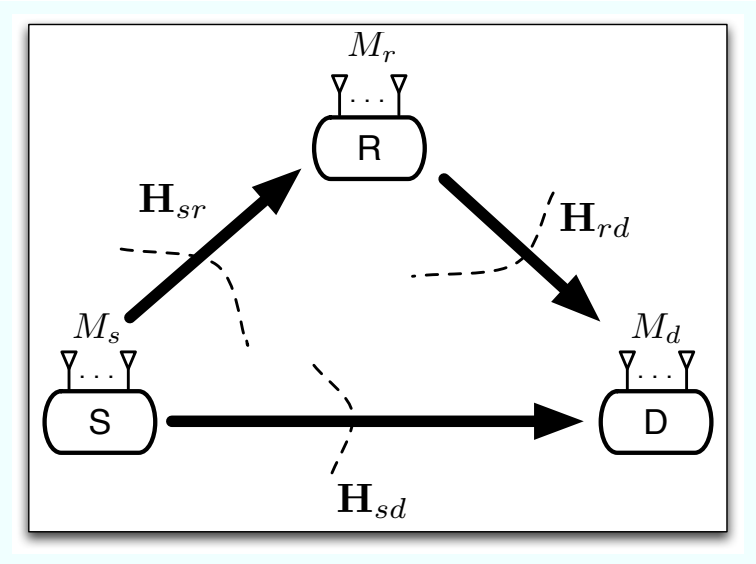

Figure 1.1 A relay network with three nodes: source, relay and destination. The respective channel matrices are denoted by $\mathbf{H}_{sr}$, $\mathbf{H}_{rd}$ and $\mathbf{H}_{sd}$.

Also, whenever dealing with point to point subsets of a cooperative network, we will consider a system with M_T transmit and M_R receive antennas. Blocks of information bits of length N_m are each encoded with a Turbo encoder with rate R. At the output of the Turbo encoder, every $\log w$-length bit sequence is mapped to one of the modulation symbols of a complex-valued constellation Ω of the order $w = |\Omega|$. The modulation symbols are multiplexed across the M_T transmit antennas and form the transmit vector $\mathbf{x} = [x_1, x_2, ..., x_{M_T}]^T$. The input-output channel model is captured by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1.11}$$

where $\mathbf{H}$ is the complex-valued $M_R \times M_T$ channel matrix, $\mathbf{n}$ is the circularly symmetric complex additive white Gaussian noise vector of size M_R and $\mathbf{y} = [y_1, y_2, ..., y_{M_R}]^T$ is the M_R-element received vector.

Note that throughout this thesis, we will be focusing on methodologies and techniques that ensure backward compatibility with the current wireless infrastructure physical layer. Therefore, we focus on using the common modulation and coding schemes. Moreover, the two time slot cooperation scenario can still take advantage of scheduled access MAC layer protocols. In other words, scheduling algorithms can be used to schedule the transmission between the source(s) and the relay(s) to ensure that both source and relay get proper time interval to transmit. However, random access MAC layer protocols can not be readily generalized to support the two time

slot scheme.

1.2 Thesis Contributions

The contributions of this thesis are three-fold:

Complexity Reduction (Chapter 3): In this thesis, we will present complexity reduction techniques to reduce the complexity of MIMO systems. In Chapter 3, we will propose different reduced-complexity MIMO detectors that could be used in the different nodes of wireless cooperative systems.

Flexible and Configurable Architectures (Chapter 4): We will also present architectures for flexible and configurable MIMO detectors. These architectures could support a range of data rates, modulation orders and numbers of antennas, and therefore, are crucial in the different nodes of cooperative systems. For instance, the relay nodes could use these architectures for supporting different transmission modes. As one example of such architectures, we present the FPGA implementation of a novel configurable and flexible sphere detector called Flex-Sphere which supports three commonly used modulation schemes, $4, 16, 64$-QAM, as well as a combination of $2, 3$ and 4 user/antenna configuration. The detector provides a data rate of up to 849.9 Mbps. The breadth-first search employed in our realization presents a large opportunity to exploit the parallelism of the FPGA in order to achieve high data rates. Algorithmic modifications to address potential sequential bottlenecks in the traditional bread-first search-based SD are highlighted in the thesis.

Cooperation (Chapters 5 and 6): In order to address the aforementioned challenges, we propose a novel Cooperative Partial Detection (CPD) approach in MIMO relay channels, where instead of applying the conventional full detection in the relay, the relay performs a partial detection and forwards the detected parts of the message to the destination. Moreover, instead of making the impractical assumption of complete channel state information in all the nodes, the proposed cooperative partial detection strategy assumes that in each communication link, channel knowledge is available only at the receiver of that link. Detecting and transmitting a subset of the source streams from the relay reduces the interference due to the relay in the second wireless network.

We will define *expansion factor*, ef, as the parameter that captures the number of streams of data detected in the relay and transmitted from the relay to the destination. Using the ef parameter, we will show that this cooperative detection scheme improves the error performance compared to non-relay scenarios with limited computational overhead in the relay. We will show that this technique can help in distributing the detection process between the relay and destination. Furthermore, the ef parameter provides the means for the relay so that it could choose, depending on its resource availability, how much of its processing power it should dedicate to helping the direct source-destination link.

Therefore, we propose a *partial* sphere detection scheme, which is designed and

proposed based on the practical limitations of wireless devices. This detection scheme is used in the relay for partial detection. We also propose a detection scheme in the destination that is based on maximal ratio combining of the received data.

It is important to note that our proposed cooperative detection scheme can be applied to a wide variety of wireless communications systems. For instance, in the context of uplink scenarios, this scheme can be applied in the MIMO terminal transmitting its spatially multiplexed signals to the basestation. Also, this scheme may be used in assisting the basestation in uplink *multi-user* detection scenarios, where multiple users with a smaller number of antennas try to use the same channel for sending the data to the basestation. As for the downlink, the MIMO relay can be used for communicating data from the basestation to terminals with multiple antennas. Note that multiple antenna mobile nodes have been discussed and proposed for IEEE 802.16 [37] and IMT-advanced [3] standards and also for 3GPP LTE [38]. In all such scenarios, the relay node can be either a dedicated MIMO relay, or another idle MIMO user.

We will discuss the complexity implications of this method, and more importantly, present hardware verification and over-the-air experimentation of CPD using the Wireless Open-access Research Platform (WARP) in Chapter 6.

Insights for Studying the Cooperation Space

While in this thesis we will focus on specific scenarios of single-relay, the insights

we obtained from these scenarios can be applied to the other cases of cooperation:

1. Partial detection/decoding with multiple relays and sources: While we focus on single-relay scenarios, the partial detection structure can be applied to multiple relays, where these multiple relays may detect different or similar subsets of the transmitted messages. The number of ways to split these subsets among the relays grows with the number of relays. Therefore, tradeoff analysis need to be performed to decide how to split these subsets among different relays.

2. Partial tree traversal: An important component of the cooperative partial detection is based on the partial traversal of the detection tree. In this thesis, we will demonstrate the performance gains and complexity reductions achieved by partial tree traversal. This technique could be applied, with modifications and adaptations, to other detection/decoding schemes with tree structures and trellises, e.g. the trellis diagram of the convolutional decoder, or the Tanner graph of LDPC decoder. For instance, the trellis diagram can be split into multiple sections or new methods for segmentations of the LDPC Tanner graph could be studied that allow partial decoding of the message.

3. Configurable architectures: an important aspect of understanding the complexity-performance tradeoff of different cooperation techniques, is whether or not such techniques can be implemented with configurable architectures. The methods that we will use in this thesis for ensuring configurability and flexibility can be extended to

other architectures.

1.3 List of Symbols and Abbreviations

Here, we provide a summary of the abbreviations and symbols used in this thesis:

CMPU (Computation Unit): A block in the sphere detection architecture to compute the PEDs,

CORDIC (Coordinate Rotation Digital Computer): A simple and efficient algorithm to calculate hyperbolic and trigonometric functions,

CPD (Cooperative Partial Detection): A technique proposed in this thesis to perform partial detection with MIMO relays,

ef (Expansion Factor): The number of detected streams in the CPD,

FDF (Full Detect and Forward Cooperation): A technique to perform full detection and forwarding in the relay

FPGA (Field Programmable Gate Array): An integrated circuit designed to be configured by the customer or designer after manufacturing, hence "field-programmable",

LDPC (Low Density Parity Check) code: A channel coding technique based on low density parity check matrices,

LLR (Log-Likelihood Ratio): Ratio that reflects the reliability of a bit being zero or one,

MIMO (Multi-input, multi-output): Multiple-antenna wireless transmitter-receiver pair,

MMSE (Minimum Mean-Squared Error): An estimation technique that minimizes the mean squared of the error,

Node Ordering Unit (NOU): A block in the sphere detection architecture to order the units,

PED (Partial Euclidean Distance): A partial distance computed during the sphere detection process,

RVD (Real-Valued Decomposition): Decomposing a complex-valued matrix into a real-valued matrix,

QAM (Quadrature Amplitude Modulation): A digital modulation scheme where groups of bits are mapped to complex symbols,

SDM (Spatial Division Multiplexing): A wireless transmission scheme where the signals are multiplexed across multiple antennas/spaces,

SNR (Signal-to-noise Ratio)

WARP (Wireless Open-access Research Platform): A wireless research platform for prototyping wireless algorithms,

ZF (Zero Forcing): A linear MIMO detection scheme,

M_T: Number of transmit antennas,

M_R: Number of receive antennas,

^{T}: Superscript denoting the transpose of a matrix,

^{H}: Superscript denoting the conjugate transpose of a matrix,

Ω: Complex-valued constellation of the modulation,

w: Order of Ω,

cov{.}: Covariance matrix function,

E{.}: Expected value,

S: Denotes the source node in the cooperative setup,

R: Denotes the relay node in the cooperative setup,

D: Denotes the destination node in the cooperative setup,

$\mathbf{H}_{sr}$: The source-relay channel matrix,

$\mathbf{H}_{rd}$: The relay-destination channel matrix,

$\mathbf{H}_{sd}$: The source-destination channel matrix.

Chapter 2
Background and Related Work

Digital communications [58] and wireless communications [28] have been ever increasing industries with significant theoretical and practical challenges for the last few decades. With the promising results of MIMO point-to-point communications [52, 42], MIMO systems have been playing a significant role in a wide variety of wireless standards, and thus, various detection algorithms have been proposed.

The sphere detection algorithm as a method of solving integer least-squares problem was proposed in [107]. Later on, this method was applied for solving MIMO detection problem in wireless systems [34, 77, 26, 46]. The K-best variation of sphere detection, which has fixed complexity, was proposed in [53, 66]. Hard-detection was further generalized to soft detection for coded systems using soft sphere detection [21, 92]. In order to further reduce the complexity of sphere detection, dynamic thresholding was proposed to control the sphere radius and reduce the number of visited nodes [60]. ASIC implementation results were presented for K-best and fixed-complexity MIMO systems [113, 76, 86, 90, 87, 85] and sphere detection [8, 93]. FPGA prototypes of sphere detection have been reported in [54, 68, 69]. Further complexity reduction techniques were presented in [67, 45, 36, 101, 29],

Trellis-based detection algorithms, which replace the tree search with trellis search,

i.e., another form of representing the tree search, were presented [99, 97] and implemented on AISC [98] and GPU [83, 80, 82, 81].

Other reduced-complexity techniques that took advantage of the breadth-first tree traversal were reported in [79, 112, 63]. Monte-Carlo Markov-Chain (MCMC) method was used to stochastically detect the transmitted signal [20, 48, 47]. Moreover, MCMC-based architecture and estimates have been presented in [65]. In the following section, we will present a brief summary of MIMO detection systems.

2.1 Review of Maximum-Likelihood Detection

The MIMO system model with M transmit antennas and N receive antennas can be described by

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2.1}$$

where $\mathbf{H}_{N\times M}$ is the channel matrix, $\mathbf{s}_{M\times 1}$ is the transmitted vector with complex elements chosen from a set of modulation constellation, $\mathbf{n}_{N\times 1}$ is the complex noise vector, and $\mathbf{y}_{N\times 1}$ is the received vector. The maximum-likelihood (ML) estimate of the transmitted signal is given by

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\Omega} \parallel \mathbf{y} - \mathbf{H}\mathbf{s} \parallel^2 \tag{2.2}$$

where Ω is the constellation set with w elements, i.e. $|\Omega| = w$, and $\parallel . \parallel^2$ denotes the ℓ^2 norm of the matrix throughout the thesis.

The ML estimate is shown to be the optimum detector in communication receivers [58]. However, as (2.2) suggests, this requires a brute-force search among all the possible candidates. In other words, for the system described above, w^M search operations are required to find the optimum solution. Thus, the complexity of maximum-likelihood (ML) increases exponentially with the number of antennas. For example, for a 4×4, 16-QAM MIMO system, 2^{16} search operations are required which considering the current VLSI area limitations is infeasible to implement.

2.2 Review of Sphere Detection

ML detectors have a high complexity in MIMO systems with high order modulation schemes and moderate number of antennas. Thus, sphere detection [107], [34] has been proposed to decrease the complexity of the search.

The norm in (2.2) can be simplified as [77]:

$$\begin{aligned} D(\mathbf{s}) &= \| \mathbf{y} - \mathbf{Hs} \|^2 \\ &= (\mathbf{y} - \mathbf{Hs})^H (\mathbf{y} - \mathbf{Hs}) \\ &= (\mathbf{y} - \mathbf{Hs})^H \mathbf{Q}\mathbf{Q}^H (\mathbf{y} - \mathbf{Hs}) \\ &= (\mathbf{Q}^H\mathbf{y} - \mathbf{Rs})^H (\mathbf{Q}^H\mathbf{y} - \mathbf{Rs}) \\ &= \| \mathbf{Q}^H\mathbf{y} - \mathbf{Rs} \|^2 \\ &= \sum_{i=M}^{1} |y_i{}' - \sum_{j=i}^{M} R_{ij} s_j|^2 \end{aligned} \tag{2.3}$$

where $\mathbf{H} = \mathbf{QR}$, $\mathbf{QQ}^H = \mathbf{I}$, $\mathbf{R}$ is an upper triangular matrix and $\mathbf{y}' = \mathbf{Q}^H\mathbf{y}$. Super-

script ^{H} denotes the conjugate transpose operator. We also define the partial distance (PD) as,

$$PD = |y_i{}' - \sum_{j=i}^{M} R_{ij} s_j|^2. \tag{2.4}$$

The summation in (2.3) can be done through a tree where the value of each node of the tree is equivalent to the partial distance of that node. This tree will have $M + 1$ levels. Moreover, each node of the tree has w children nodes where w is the number of constellation points. Furthermore, since the external summation is over non-negative terms, children nodes have partial distances greater than or equal to the partial distances of their parent.

If the search is limited to those nodes whose partial distances are smaller than a pre-specified threshold, the number of visited nodes, and hence the complexity, would decrease. In other words, imposing the condition that $D(\mathbf{s}) \leq R^2$, will lead to pruning out the nodes whose partial distances are greater than R^2. Note that whenever a node is pruned out, its children can also be pruned out. This is because of the monotonic increasing nature of partial distances.

Figure 2.1 shows a specific case of a MIMO system with four transmit antennas, each using a two-element modulation constellation, e.g. BPSK. Applying maximum-likelihood (ML) to this detection problem is equivalent to visiting all the 31 possible nodes of the search tree; whereas, imposing a threshold, i.e. a radius of 9, leads to visiting 19 nodes. The complexity reduction is more significant with more strict

thresholds and higher order modulation schemes.

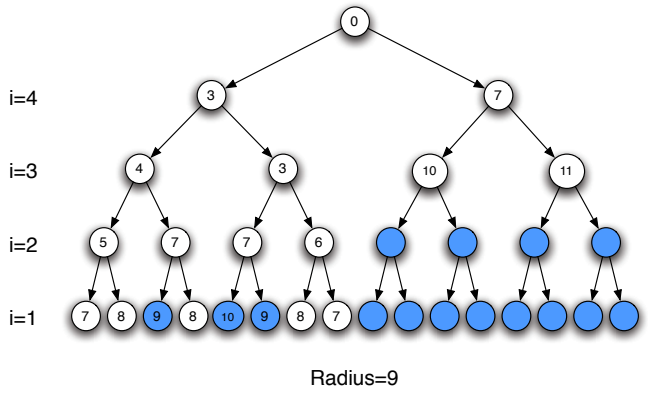

Figure 2.1 Computing partial distances using a tree. Numbers in each node indicate the partial distance

2.2.1 Channel Decomposition Techniques

An alternative technique to QR decomposition is using Cholesky factorization. Cholesky factorization is a matrix decomposition technique that decomposes the positive definite $\mathbf{H}^H\mathbf{H}$ matrix to $\mathbf{U}^H\mathbf{U}$, where $\mathbf{U}$ is an upper triangular matrix. However, as shown in [31], in order to speed up the processing rate, it is critical to avoid division and square-root operations, that are used in both typical QR and Cholesky factorization techniques. Therefore, CORDIC-based architectures have been proposed [50] for QR decomposition and systolic array architectures, using square-root free Givens rotations, have been proposed for fixed point computations [43, 84]. Note that while the square-root free Cholesky factorization, $\mathbf{H}^H\mathbf{H} = \mathbf{U}^H\mathbf{D}\mathbf{U}$, avoids the square root

operation, it introduces a diagonal matrix **D** that makes the tree traversal a less straightforward process. Thus, QR decomposition is a more architecture-friendly solution for VLSI implementation of sphere detection [96].

Moreover, there are a number of different ways to compute the QR decomposition of the **H** matrix [40]: the Gram-Schmidt method is based on iterative subtraction of the matrix columns projections. This method generally suffers from poor numerical stability due to rounding errors. Householder transformation, another common QR decomposition technique, computes the QR decomposition based on reflecting the matrix column; however, it requires both division and square root operations, which is challenging given the VLSI area constraints. Finally, the Givens rotation method is based on multiple rotations, where in each rotation, one of the sub-diagonal elements of the matrix is zeroed to form the upper triangular matrix. For the reasons discussed earlier, square-root free Givens rotation QR technique [43] better fits the architecture constraints, and hence, is most commonly used for pre-processing stage of the MIMO sphere detection.

2.3 Architecture for Sphere Detection

The norm in (2.3) can be computed in M iterations starting with $i = M$. When $i = M$, i.e. the first iteration, the initial partial norm is set to zero, $PN_{M+1} = 0$. At each iteration, partial distances, $PD_i = |y_i' - \sum_{j=i}^{M} R_{i,j} s_j|^2$ corresponding to the i-th level, are calculated and added to the partial norm of the respective parent node in

the $(i-1)$-th level, $PN_i = PN_{i-1} + PD_i$. Finishing the iterations gives the final value of the norm. One can envision this iterative algorithm as a tree traversal problem where each level of the tree represents one i value, each node has its own PN, and w children, see Figure 2.2. In order to reduce the search complexity, a threshold, C, can be set to discard the nodes with $PN > C$. Therefore, whenever a node k with a $PN_k > C$ is reached, any of its children will have $PN \geq PN_k > C$. Hence, not only the k-th node, but also its children, and all nodes lying beneath the children in the tree, can be pruned out. The tree can be traversed either vertically, known as depth-first search (DFS) [8], [60]; or level by level, called breadth-first search (BFS) [112], [66]. Our initial approach is a modified DFS-based scheme [60]. In later chapters, we will propose a breadth first architecture, also known as Flex-Sphere.

Figure 2.3 shows the main blocks of the detector. The tree traversal unit (TTU) is responsible for searching through the tree. It functions as a control unit to handle the flow of data between the other two blocks. Computation unit (CMPU) consists of parallel datapaths to compute the PNs for all the children nodes of another node. Node ordering unit (NOU) finds the minimum among all the P possible children nodes whose PNs have been calculated in the CMPU. Visiting each node in the tree is equivalent to one iteration of the {TTU, CMPU, NOU} loop.

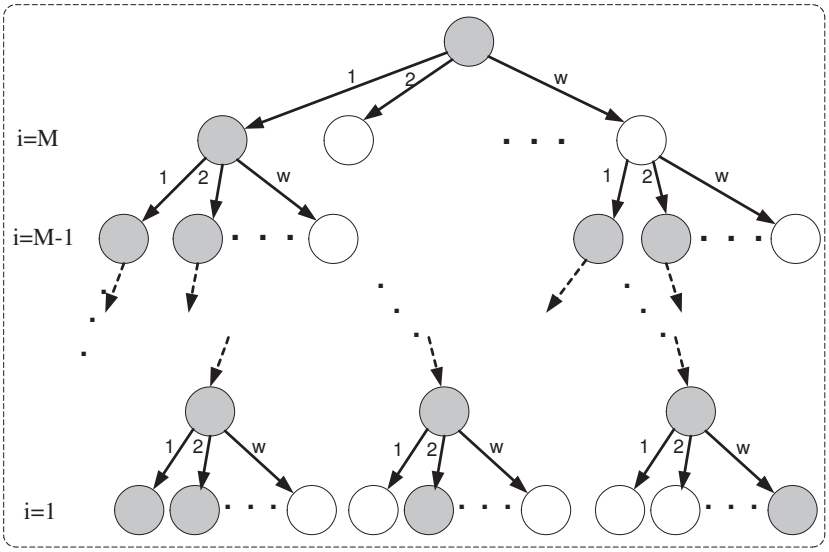

Figure 2.2 Calculating the distances using a tree. Partial norms, PNs, of dark nodes are less than the threshold. White nodes are pruned out.

Computation Unit (CMPU)

Computing the PDs for all the children of each node can be quite resource and cycle consuming.

$$PD = |y_i' - \sum_{j=i}^{M} R_{i,j} s_j|^2 \tag{2.5}$$

$$= |y_i' - R_{i,i}s_i - R_{i,i+1}s_{i+1} - ... - R_{i,M}s_M|^2 \tag{2.6}$$

for all the complex $s_i \in \Omega$. Once the PDs are calculated, they are added to the partial norm of their parent node to form their own partial norm, $PN_i = PN_{i-1} + PD_i$.

There are some points to consider in (2.6). While s_i needs to take all the different constellation points; $s_{i+1}, ..., s_M$ are fixed, and have the same value at that specific

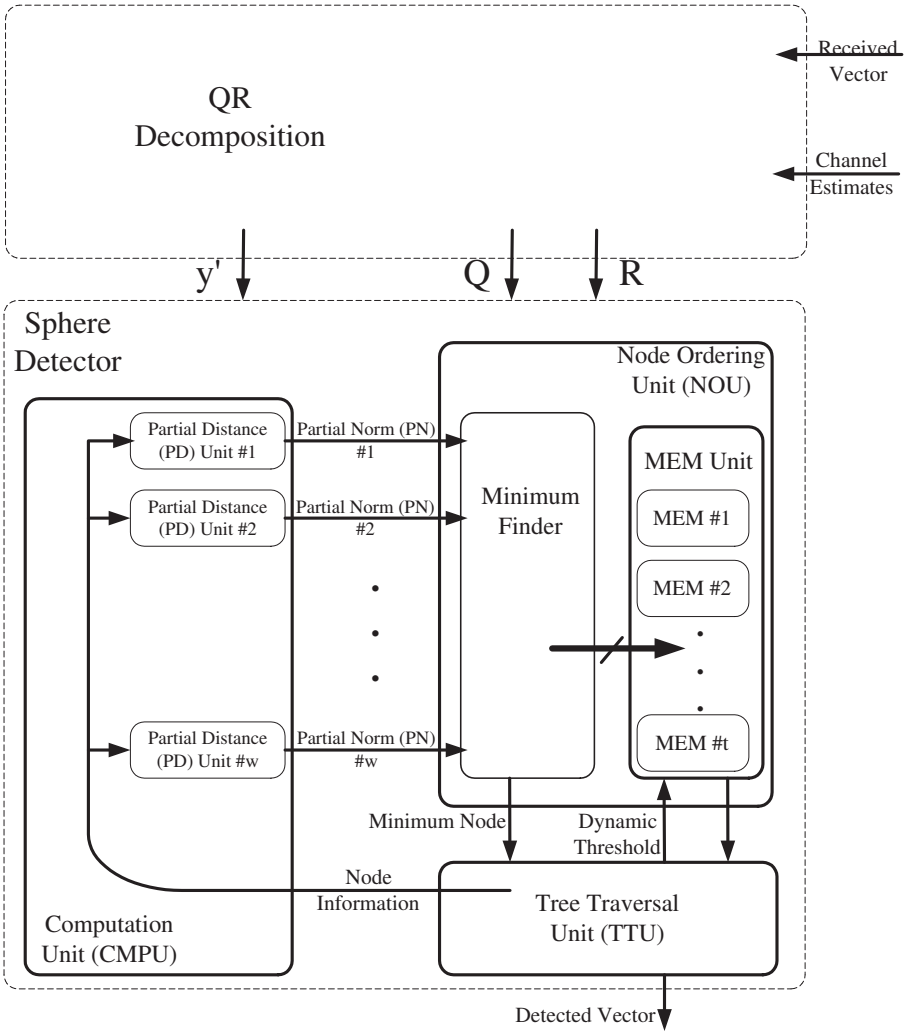

Figure 2.3 Sphere detection block diagram. Note three subblocks, TTU, CMPU, NOU, forming the loop.

level of computation, hence can be computed only once. The symbols s_k, for $k = 1, ..., M$, are chosen from a *complex* constellation Ω, and the number of elements in Ω, i.e. modulation order, is w. Also note that $diag(\mathbf{R})$ are real numbers, and all the other off-diagonal terms in the upper triangle of $\mathbf{R}$ are complex numbers. Assuming four real multipliers and two real adders for each complex multiplier and two real adders for each complex adder, the overall number of real multipliers and adders for a CMPU is given in Table 2.1. Since the same CMPU block is used for different levels of the tree, i.e. different i, and also different antennas can use different modulations, we need to design it for the worst case. Hence, a trivial architecture for CMPU will have $(4(M-1)+4w)$ real multipliers and $(2(2M-1)+2w)$ real adders.

Table 2.1 Initial number of multipliers and adders for CMPU

Step	Operation	Real MUL	Real Adder
1	$R_{i,i}s_i$	$2w$	0
2	$\sum_{j=i+1}^{M} R_{i,j}s_j$	$4(M-i)$	$2(M-i)$
3	$y_i{}' - R_{i,i}s_i - \sum_{j=i+1}^{M} R_{i,j}s_j$	0	$2(M-i+1)$
4	$\lvert y_i{}' - R_{i,i}s_i - \sum_{j=i+1}^{M} R_{i,j}s_j\rvert^2$	$2w$	w
5	$PN_i = PN_{i-1} + PD_i$	0	w
	Total	$4(w+M-i)$	$4(M-i)+2+2w$

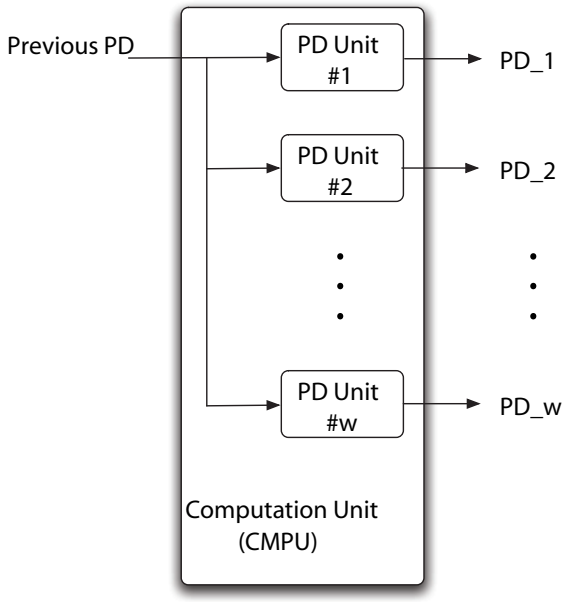

Figure 2.4 Computation Unit (CMPU)

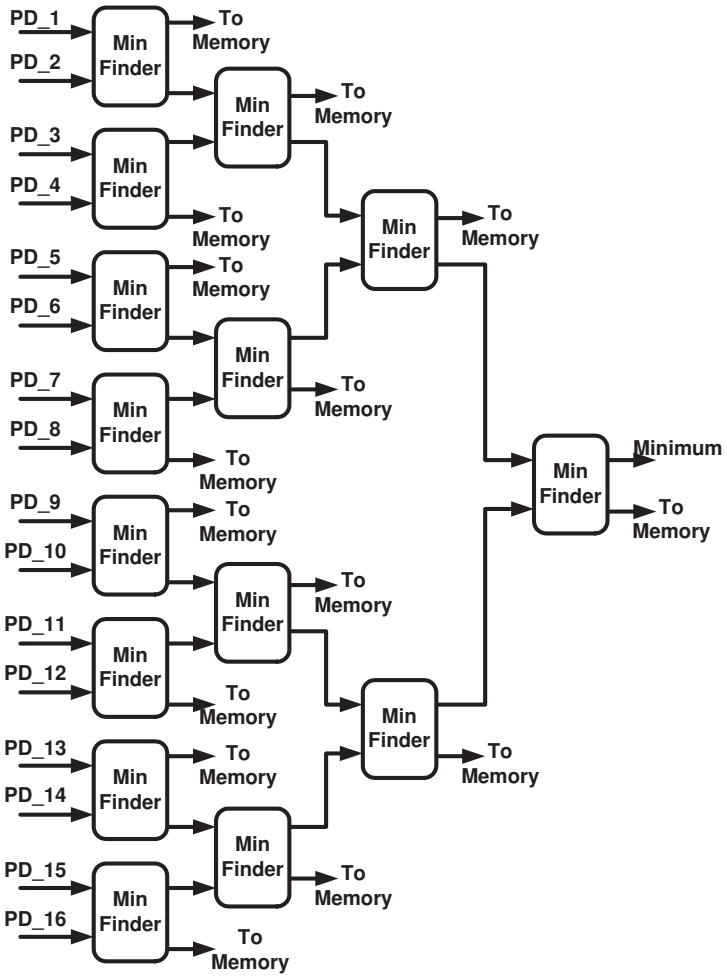

Figure 2.5 Node Ordering Unit (NOU). Each Min Finder block finds the minimum of its two inputs, and passes that minimum to the next minimum finder. The larger output of each Min Finder block will be saved into memory only if it is inside the local threshold.

The number of multipliers can be considerably reduced by noting the fact that most of the multiplicands are constellation points with their real and imaginary values taken from a small set of integer numbers. Therefore, each of the real multipliers used to form a complex multiplier can be replaced with a combination of adders, shifters and multiplexers. It can be verified that using this property of the multiplicands, the CMPU needs only $(2w)$ real multipliers; the rest of the multipliers in the original CMPU are replaced by different adders and shifters resulting in an overall number of $(2M-6)\sqrt{w}+4w+2$ adders and $(4(M-1)\log_2(\sqrt{w}))$ multiplexers. Using synthesis results, Table 2.2 compares the resources required for the original CMPU and the modified CMPU based on multiplier reduction. Figure 2.6 shows how such real multipliers can be implemented using adders/shfiters.

Table 2.2 Required arithmetic units for two different CMPU implementations with $M = 4$

Modulation (w)	Original CMPU		Modified CMPU			Approximate Area Reduction
	Real MUL	Real Adder	Real MUL	Real Adder	Real MUX	
16-QAM	76	46	32	74	24	51.1%
64-QAM	268	142	128	274	36	45.8%
256-QAM	1036	526	512	1058	48	44.6%

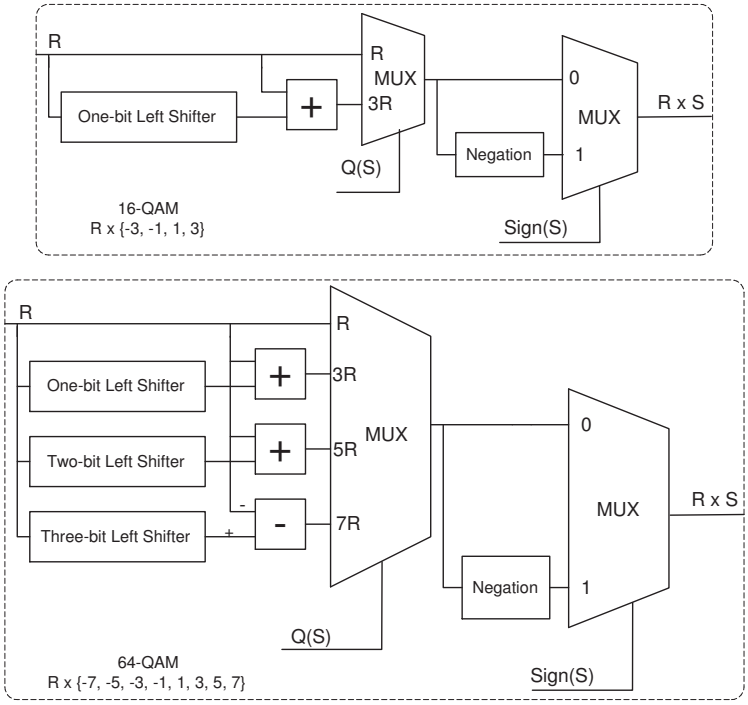

Figure 2.6 Reduced Complexity Multiplier Architecture for 64-QAM. $Q(.)$ maps the value of S to proper MUX indices. Similar combination of adder-shifters can be used for higher order modulations. These real multipliers can be used to perform complex multiplications, where a complex multiplier corresponds to four real multipliers and two real adders.

There is no data dependency between steps 1 and 2 in Table 2.1. Hence, they can be performed in parallel. The operations listed in Table 2.1, except the second step which is common for all the children, are repeated in all of the partial distance units. Changing the modulation order, i.e. w, only modifies the number of parallel partial distance units. Hence, for a system supporting different modulation schemes, it is sufficient to design the CMPU for the largest modulation order, and it can support other modulations.

Node Ordering Unit (NOU)

The children nodes need to be compared with the dynamic threshold, C_i. If outside the dynamic sphere, they should be discarded; otherwise, kept for further computations. Among those kept candidates, the best one should be sent for the next tree level of the computations in the CMPU, and the rest will be saved in memory. Unlike the K-best approach, where the structure is based on *sorting* the candidates, it is not necessary to perform sorting in DFS-based schemes. However, we have included a minimum finder in the NOU to find the minimum partial norm, minimum-PN, among all the w different PNs generated in the CMPU. Notice that continuing with the minimum-PN, results in reaching smaller norm leafs. The global threshold, C, is updated with the norm of a leaf whenever any leaf is reached. Therefore, the concept of continuing with the minimum-PN node greatly reduces the threshold [60] [8].

The minimum finder requires $w - 1$ compare-select blocks searching among the

possible candidates in a $(\log_2 w)$-level tree. The best candidate, i.e. the minimum, is sent to the tree traversal unit (TTU), and the rest are saved in the memory as long as their partial norms are less than the dynamic threshold of that level. The size of the MEM unit is very small as the dynamic threshold updating scheme prunes out considerable number of nodes during the search process, see Figure 2.7.

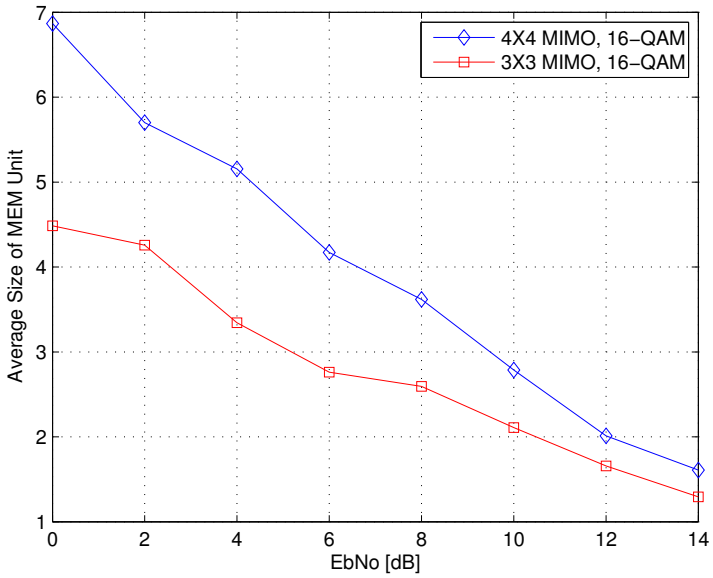

Figure 2.7 Average memory size for the dynamic threshold updating.

Since the data needs to be compared and listed in a queue to be saved in the memory, higher modulation orders mean longer queues and longer read-write time from MEM unit. Therefore, memory unit interface can become a major bottleneck that reduces the data rate for higher order modulations. In order to avoid this, and

keep the architecture easily scalable without throughput penalty, we propose using separate memory modules that can be accessed simultaneously, so that the average time required to save all of them in the MEM unit is essentially divided by the number of memory modules. If the number of clock cycles for writing the remaining PNs into the MEM unit is C_{mem}, then $t = (w-1)/C_{mem}$ memory modules are used in the MEM unit. The optimum timing for saving into the MEM and avoid stalling, is to do that while other blocks, i.e. CMPU and TTU, are processing the data. Hence, a reasonable choice for C_{mem} is $C_{mem} = C_{TTU} + C_{CMPU}$. Thus, the number of memory modules in the MEM unit is, $t = (w-1)/(C_{TTU} + C_{CMPU})$. Note that using this architecture, the transfer time between the MEM unit and other blocks do not increase as higher modulations are utilized.

Tree Traversal Unit (TTU)

The TTU handles the flow of data between the CMPU and NOU. Computation of the current threshold, C_i, can be the updated radii, or can be done based on:

$$R_i = \frac{R(M+1-i)}{M}. \tag{2.7}$$

where M is the number of transmit antennas, R_i is the radius at the i-th level, and R is the radius at last level.

The dynamic threshold, C_i, is chosen from the set $\{C/M, 2C/M, ..., (M-1)C/M\}$. Similar to CMPU, no explicit multiplier is required to compute C_i since all those

integer multiplications can be performed using only adders and shifters.

Throughput

Table 2.3 gives the number of cycles required to generate outputs in each of the blocks. In order to guarantee high clock frequencies, the CMPU block, which goes through the steps in Table 2.1, has been heavily pipelined. Moreover, the NOU is pipelined in such a way that every two sequentially successive compare-select blocks in the tree structure form one pipeline stage. The TTU needs one clock cycle in the case that the MIN output from the NOU is not a leaf of the tree and is inside the dynamic threshold. If not, C_{mem} extra clock cycles are required to read the data from the memory. The last row of the table shows the overall number of cycles required to do one iteration, i.e. visit one node. $E\{C_{TTU}\}$, the expected value of the number of cycles of TTU, captures the uncertainty in the number of cycles of the TTU unit.

Table 2.3 Number of clock cycles required to perform each step

Unit Name	Number of Clock Cycles
C_{CMPU}	5
C_{NOU}	$\lceil \frac{1}{2} \log_2 w \rceil$
C_{TTU}	1 or $1 + C_{mem}$
$\sum\{C_{TTU}, C_{CMPU}, C_{NOU}\}$	$5 + \lceil \frac{1}{2} \log_2 w \rceil + E\{C_{TTU}\}$

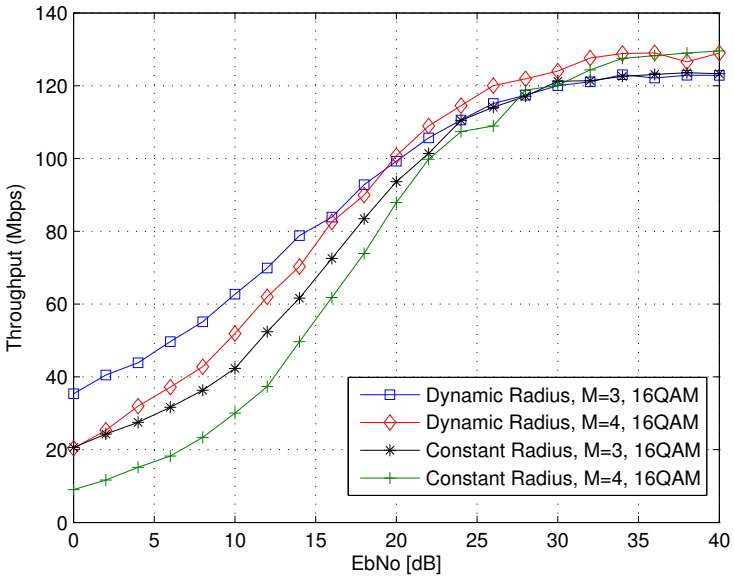

Figure 2.8 Throughput of the architecture for $f_{max} = 300$ MHz, and number of transmit antennas, M, of 3 and 4.

If N_v nodes are visited in the tree to find the detection solution, then the throughput can be calculated based on:

$$Throughput = \frac{M(\log_2 w) f_{max}}{N_v(5 + \lceil \frac{1}{2} \log_2 w \rceil + E\{C_{TTU}\})}. \tag{2.8}$$

Note that N_v highly depends on the radius reduction scheme, whether we use constant threshold, C, which is only updated with new leafs, or dynamic threshold, C_i, given in (2.7). Figure 2.8 compares the throughput for different dynamic and constant radius examples.

While the architecture discussed so far utilizes depth-first search, other tree traver-

sal techniques can also be used to perform MIMO detection. In the next chapters, we will present flexible and configurable architectures that take advantage of breadth-first style tree traversal to perform MIMO detection in cooperative systems with relays.

2.4 Cooperative Communications Using MIMO Nodes

Cooperative communications, and in particular relay channels, were originally introduced and studied in [32, 33] where lower and upper bounds on the capacity of relay channels were derived [103], which were later improved and extended for half-duplex scenarios [11, 9]. User cooperation reemerged again in [16, 17, 14] as a form of diversity in uplink scenarios. Different relaying protocols and relay selection schemes were studied and compared in [55, 56, 57, 27, 15, 18, 91, 106]. Moreover, a new cut-set theorem, power control strategies and LDPC relay code designs were proposed [70, 71].

In order to facilitate user cooperation in practical scenarios, coded cooperation was proposed in [100, 72, 102, 51], and in particular, LDPC code design methodologies have been proposed for relay channels in [12, 13, 10]. Furthermore, in order to reduce the overhead of decoding in the relay, various distributed decoding schemes have been proposed in [74, 73], where the relay performs a partial decoding as opposed to the conventional full-decoding of the message.

Partial decode and forward was introduced and studied in [103, 49], where the relay only decodes part of the message and forwards that to the destination, and is

derived from block-Markov coding. A relay rate-constrained cooperative system was also proposed and studied in [95], where cut-set upper bounds and rate-constrained cooperative schemes were derived. Another partial decode-and-forward strategy was proposed in [111], which applies broadcast strategy to fading relay channels, and relies on a two-level superposition coding strategy. The source uses a layered coding, and the relay decodes either the first layer during weak channel states, or both of the layers if the channel state is "good". In this thesis, we are focusing on using the modulation/coding schemes that have been used in the standards to maintain backward compatibility with the currently in-place wireless infrastructures. Using the currently available modulation and coding physical layer features, therefore, allows for direct application of the current wireless nodes in source-relay settings.

More recently, there have been some attempts to study the theoretical benefits and bounds on deploying MIMO nodes in cooperative scenarios, both as relays and as source/destination pairs. In doing so, lower bounds and upper bounds for MIMO relay networks were given in [23, 24], and capacity scaling factors were derived for multi-hop MIMO relays [44] and two-way relaying schemes [94]. Optimal precoder designs for MIMO relays were discussed in [105], and achievable sum rates for multiuser MIMO relay channels were proposed in [104, 39]. In [24], full-duplex MIMO relay channels are studied and using message splitting and partial cooperation, rate bounds are derived. There have been efforts recently to implement and prototype cooperative

systems [110, 75, 41, 89], and an OFDM transceiver design was presented in [88] using the WARP platform.

Chapter 3
Hardware Architecture and Implementation for MIMO Detection

In this section, we will discuss further complexity reduction techniques in MIMO systems.

3.1 Soft MMSE Receiver

In this section, we present the soft MMSE (S-MMSE) receiver. As shown in [22], in order to detect the x_j symbol, $j = 1, ..., M_T$, the expected values of the transmitted symbols are computed using the LLR values, L_C, from the channel decoder:

$$\begin{aligned} \tilde{x}_j &= \sum_{\tilde{x} \in \Omega} \tilde{x} P(x_j = \tilde{x}) \\ &= \sum_{\tilde{x} \in \Omega} \tilde{x} \prod_{l=1}^{\log w} [1 + \exp(-\{\tilde{x}\}_l \cdot L_C(b_{l,j})]^{-1}, \end{aligned} \tag{3.1}$$

where $b_{l,j}$ is the l-th bit in the x_j symbol. Note that for the first outer iteration, the vector L_C's are all equal to zero since no channel decoding has been done yet.

The vector of these mean values are formed, while replacing the current symbol with zero:

$$\tilde{\mathbf{x}}_j = [\tilde{x}_1, ..., \tilde{x}_{j-1}, 0, \tilde{x}_{j+1}, ..., \tilde{x}_{M_T}]^T. \tag{3.2}$$

Performing a soft cancellation using the vector of Eq. (3.2) is then done so that

the effect of the other modulation symbols are cancelled:

$$\tilde{\mathbf{y}}_j = \mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}_j \tag{3.3}$$

The MMSE filter is applied then to obtain:

$$\tilde{z}_j = \mathbf{w}_j^H \tilde{\mathbf{y}}_j \tag{3.4}$$

where, as shown in [22], the filter coefficient is computed according to

$$\mathbf{w}_j = (\mathbf{H}\Delta_j\mathbf{H}^H + \mathbf{I})^{-1}\mathbf{H}\mathbf{e} \tag{3.5}$$

where

$$\Delta_j = \mathrm{cov}\{\mathbf{x}_j - \tilde{\mathbf{x}}_j\} \tag{3.6}$$

is the covariance matrix of the $\mathbf{x}_j - \tilde{\mathbf{x}}_j$ random vector.

Finally, the output LLR values, L_M, are computed according to

$$\begin{aligned} L_M(b_i) &= \log\frac{P(b_i=+1|z_j)}{P(b_i=-1|z_j)} - \log\frac{P(b_i=+1)}{P(b_i=-1)} \\ &= \log\frac{\displaystyle\sum_{x^+\in\mathcal{S}_{i,j}^+} \exp\left(\frac{-\|z_j-\mu_j x^+\|^2}{\eta_j^2} + \sum_{k=1,k\neq i}^{\log w}\{x_j^+\}_k\cdot\frac{L_C(b_k)}{2}\right)}{\displaystyle\sum_{x^-\in\mathcal{S}_{i,j}^-} \exp\left(\frac{-\|z_j-\mu_j x^-\|^2}{\eta_j^2} + \sum_{k=1,k\neq i}^{\log w}\{x_j^-\}_k\cdot\frac{L_C(b_k)}{2}\right)}, \end{aligned} \tag{3.7}$$

where

$$\mu_j = E\{z_j|x_j\} \tag{3.8}$$

$$\eta_j^2 = \mathrm{var}\{z_j\} = \mu_j - \mu_j^2 \tag{3.9}$$

are the mean and variance of z_j.

The L_M LLRs are then passed to the Turbo decoder to perform the channel decoding and generate the L_C for the next outer iteration. A similar procedure is repeated for I outer iterations, after which the decoder output is considered the final decoded bits.

Figure 3.1 illustrates the architecture for the MMSE receiver.

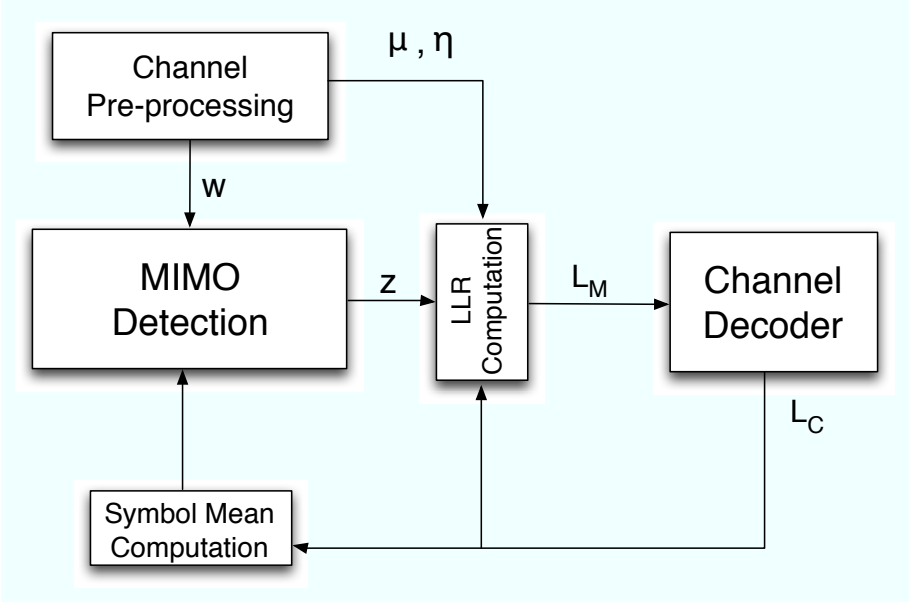

Figure 3.1 The architecture for the reduced complexity soft MMSE receiver.

3.2 Reduced Complexity Soft MMSE Receiver

In this section, we propose three techniques to reduce the complexity of the soft MMSE receiver, i.e., Reduced Complexity Soft MMSE Receiver (RC-MMSE).

3.2.1 Covariance Matrix Δ_j

The covariance matrix Δ_j has to be updated in (3.6) for every new iteration. Therefore, the MMSE weights $\mathbf{w}_j$ need to be re-computed for every new iteration. In [78], the authors have proposed using fixed covariance matrix for single antenna single carrier systems. In order to reduce the complexity of computing the MMSE weights, we propose using the first iteration covariance matrix throughout the I iterations, i.e. $\Delta_j = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

3.2.2 Max-log Approximation

In order to avoid using the look-up tables for computing the exp functions of (3.7), we propose using the Max-log approximation [21]:

$$L_M(b_i) \simeq \frac{1}{2} \max_{x^+ \in \mathcal{S}_{i,j}^+} \left(\frac{-\|z_j - \mu_j x^+\|^2}{\eta_j^2} + \sum_{k=1, k\neq i}^{\log w} \{x_j^+\}_k \cdot \frac{L_C(b_k)}{2} \right)$$
$$- \frac{1}{2} \max_{x^- \in \mathcal{S}_{i,j}^-} \left(\frac{-\|z_j - \mu_j x^-\|^2}{\eta_j^2} + \sum_{k=1, k\neq i}^{\log w} \{x_j^-\}_k \cdot \frac{L_C(b_k)}{2} \right). \tag{3.10}$$

3.2.3 Using ℓ^1 norm

Using the squared form of the norm $-\|z_j - \mu_j x^-\|^2$ in (3.10) requires performing several multiplications or squaring operations for every single bit. Therefore, we

Table 3.1 Complexity Count of Different Operations in MMSE Receiver.

	Multiplication	Addition and Compare-Select
$C_{CPP}(M_T)$	$\gamma \cdot (5.5M_T^3 + 2M_T^2 + 0.5M_T + 4M_T + 4M_T^2)$	$\theta \cdot \left(5M_T + 5M_T^2\right)$
$C_{MIMO}(M_T)$	$\gamma \cdot (4M_T\,(M_T\,(M_T - 1) + M_T))$	$\theta \cdot \left(2\,(M_T - 1) + 5\left(M_T - 1 + M_T^2\right)\right)$
$C_{SMC}(M_T, w)$	$\gamma \cdot (M_T\,((w \log w) + 2w))$	$\theta \cdot (M_T\,((w \log w) + 2w))$
$C_{LLR}(M_T, w, I)$	$\gamma \cdot (M_T w \log w)$	$\theta \cdot (2M_T w \log w + \text{step}(I-1)\,\{(M_T\,(w-1)\log w)\})$
$C_{Decoder}(T_I, N_m)$	-	$\theta \cdot (44T_I N_m) + \beta \cdot (39T_I N_m)$

propose using the ℓ^1 norm, i.e. $-\|z_j - \mu_j x^-\|$, instead.

3.3 Complexity Analysis and Hardware Architecture

Keeping in mind that each complex multiplier corresponds to four real-valued multipliers and two real-valued adders, and that every complex adder corresponds to two real-valued adders, the complexity of different units of Figure 3.1 is given in Table 3.1, where step$(I-1)$ is used to ensure the last set of computation is done for outer iterations $I > 1$, and is equal to:

$$\text{step}(t) = \begin{cases} 1, & t \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

Moreover, we use θ, β and γ to represent the hardware-oriented costs for one adder, one compare-select and one multiplication operation, respectively. Based on FPGA and ASIC estimates, we choose $\theta = 1$, $\beta = 1$ and $\gamma = 10$ throughout this chapter. The number of inner Turbo decoder iterations is denoted by T_I and the information bit sequence length is N_m. Also, in order to compute the resources required to perform

the QR decomposition in the Channel Pre-processing Unit, we assumed the modified Gram-Schmidt QR decomposition.

Therefore, the total computation is given by

$$\begin{aligned} C_{Total} &= \frac{N_m}{R \cdot M_T \log w} \cdot \{C_{CPP}(M_T) \\ &+ I \cdot C_{MIMO}(M_T) + (I-1) \cdot C_{SMC}(M_T, w) \\ &+ C_{LLR}(M_T, w, I) + (I-1) C_{LLR}(M_T, w, I)\} \\ &+ C_{Decoder}(T_I, N_m) \end{aligned} \tag{3.12}$$

Figure 3.2 compares the complexity of the conventional soft MMSE receiver of section 3.1 with the RC-MMSE receiver of section 3.2 for 4 transmit antennas and different signal modulations.

3.4 Simulation Results

In this section, we present the Bit Error Rate (BER) simulation results for a 4×4 system using both the conventional soft MMSE receiver and the proposed RC-MMSE receiver. We assume i.i.d Rayleigh fading channels.

Figure 3.3 shows the BER performance using the conventional MMSE receiver of section 3.1. The number of transmit and receive antennas are equal to 4, and the 4-QAM and 16-QAM modulations are assumed. Figure 3.4 shows the BER performance for a similar transmission system with the RC-MMSE receiver. Note that in the case of 4-QAM modulation, there is between 0.5 and 1 dB BER performance loss at

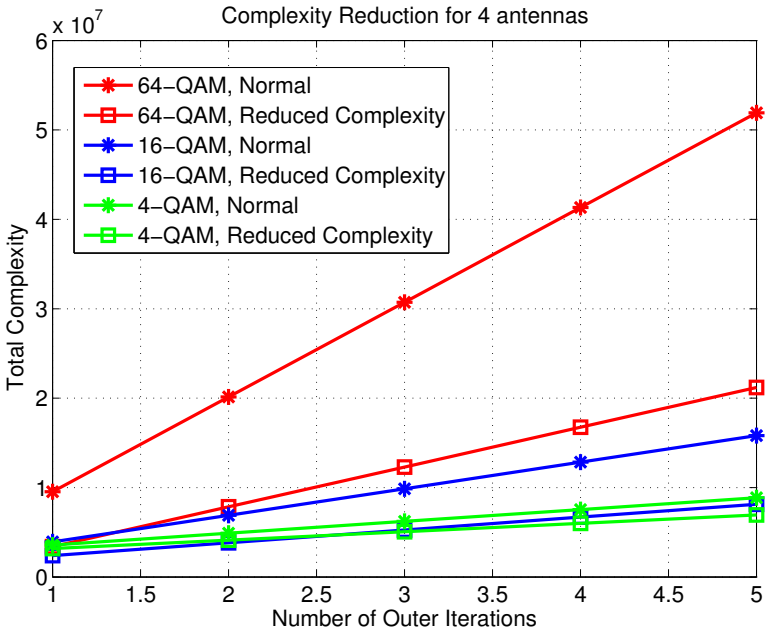

Figure 3.2 The total computation count for a 4×4, $\{4, 16, 64\}$-QAM system for different number of outer iterations.

BER = 10^{-4} for different outer iterations, and in the case of 16-QAM modulation, the BER performance loss is between 1 and 1.5 dB.

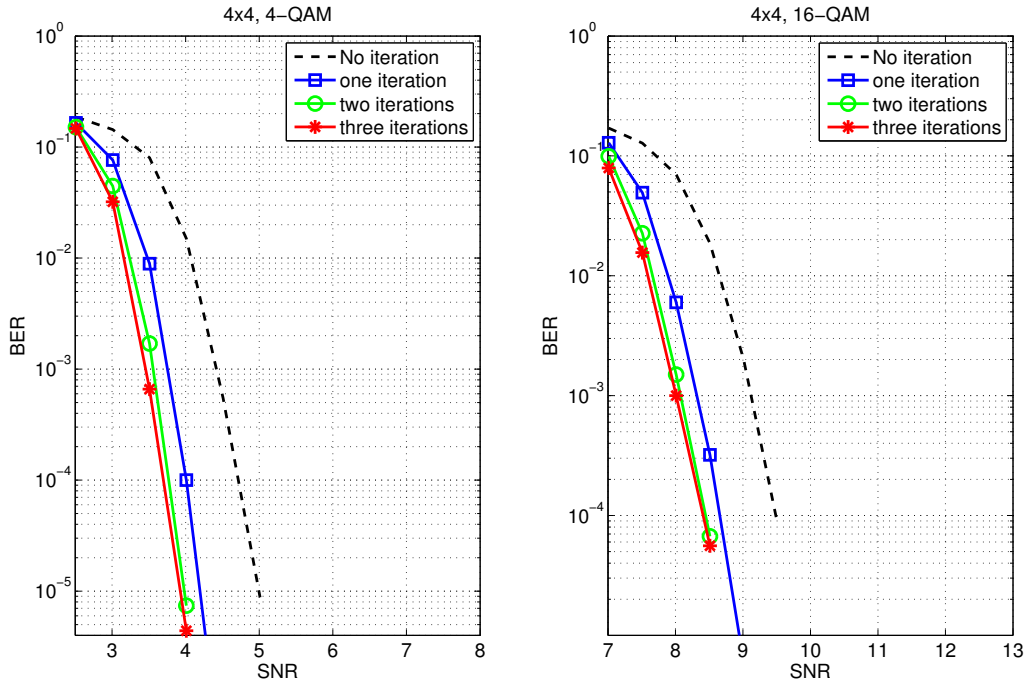

Figure 3.3 The BER performance for a 4×4, $\{4, 16\}$-QAM system using the soft MMSE receiver.

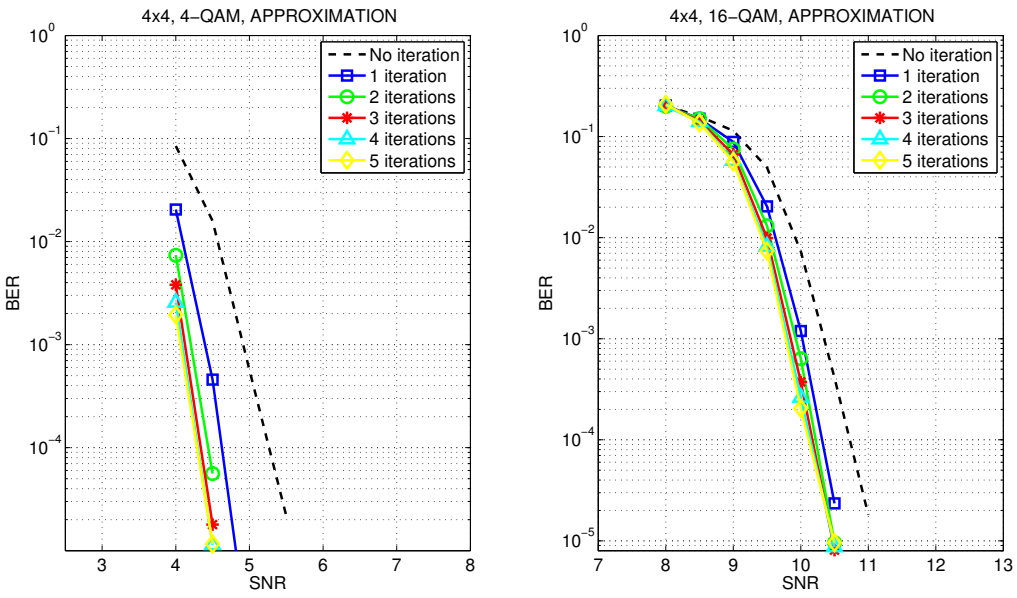

Figure 3.4 The BER performance for a 4×4, $\{4, 16\}$-QAM system using the Reduced Complexity MMSE receiver.

Chapter 4
Design and Implementation of a Flexible Detector for the Relay and Destination

The complex-valued MIMO equation can be decomposed into real-valued numbers as follows [112]:

$$\mathbf{y} = \mathbf{Hs} + \mathbf{n} \tag{4.1}$$

corresponding to

$$\begin{pmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{pmatrix} = \begin{pmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{pmatrix} \begin{pmatrix} \Re(\tilde{\mathbf{s}}) \\ \Im(\tilde{\mathbf{s}}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{\mathbf{n}}) \\ \Im(\tilde{\mathbf{n}}) \end{pmatrix} \tag{4.2}$$

with $M = 2M_T$ and $N = 2M_R$ presenting the dimensions of the new model.

We call the ordering in (4.1) the conventional ordering. Using the conventional ordering, all the computations can be performed in real arithmetic, which would simplify the implementation complexity. Note that after the real-valued decomposition, each $s_i, i = 1, ..., M$, in $\mathbf{s}$ is chosen from a set of real numbers, Ω'_i, with $w_i = \sqrt{w'_i}$ elements. For instance, for a 64-QAM modulation, each s_i can take any of the values in the set $\Omega' = \{\pm 7, \pm 5, \pm 3, \pm 1\}$.

The general optimum detector for such a system is the maximum-likelihood (ML) detector which minimizes $\| \mathbf{y} - \mathbf{Hs} \|^2$ over all the possible combinations of the $\mathbf{s}$

vector. Notice that for high order modulations and a large number of antennas, this detection scheme incurs an exhaustive exponentially growing search among all the candidates, and is not practically feasible in a MIMO receiver. However, it is shown that using the QR decomposition of the channel matrix, the distance norm can be simplified [77] as follows:

$$\begin{aligned} D(\mathbf{s}) &= \| \mathbf{y} - \mathbf{Hs} \|^2 \\ &= \| \mathbf{Q^H y} - \mathbf{Rs} \|^2 = \sum_{i=M}^{1} |y_i{}' - \sum_{j=i}^{M} R_{i,j} s_j|^2 \end{aligned} \tag{4.3}$$

where $\mathbf{H} = \mathbf{QR}$, $\mathbf{QQ^H} = \mathbf{I}$ and $\mathbf{y}' = \mathbf{Q^H y}$. Note that the transition in (4.3) is possible through the fact that $\mathbf{R}$ is an upper triangular matrix.

The norm in (4.3) can be computed in $M = 2M_T$ iterations starting with $i = M$. When $i = M$, i.e. the first iteration, the initial partial norm is set to zero, $T_{M+1}(\mathbf{s}^{(M+1)}) = 0$. Using the notation of [8], at each iteration the Partial Euclidean Distances (PEDs) at the next levels are given by

$$T_i(\mathbf{s}^{(i)}) = T_{i+1}(\mathbf{s}^{(i+1)}) + |e_i(\mathbf{s}^{(i)})|^2 \tag{4.4}$$

with $\mathbf{s}^{(i)} = [s_i, s_{i+1}, ..., s_M]^T$, and $i = M, M-1, ..., 1$, where

$$|e_i(\mathbf{s}^{(i)})|^2 = |y_i{}' - R_{i,i} s_i - \sum_{j=i+1}^{M} R_{i,j} s_j|^2 \tag{4.5}$$

$$= |b_{i+1}(\mathbf{s}^{(i+1)}) - R_{i,i} s_i|^2. \tag{4.6}$$

One can envision this iterative algorithm as a tree traversal with each level of the tree corresponding to one i value, and each node having w_i' children.

The tree traversal can be performed in both depth-first and breadth-first manners. The depth-first computation presents a variable complexity, which makes it hard to deploy in constrained scenarios. In the K-best method, which is the most common way of performing breadth-first search, at each level, only the best K nodes, i.e. the K nodes with the smallest T_i, are chosen for expansion. This type of detector is generally known as the K-best detector. Note that such a detector requires sorting a list of size $K \times w'$ to find the best K candidates. For instance, for a 16-QAM system with $K = 10$, this requires sorting a list of size $K \times w' = 10 \times 4 = 40$ at most of the tree levels. This introduces a long delay for the next processing block in the detector unless a highly parallel sorter is used. Highly parallel sorters, on the other hand, consist of a large number of compare-select blocks, and result in dramatic area increase.

4.1 Flex-Sphere SDM/SDMA Detector

In order to simplify the sorting step, which significantly reduces the delay of the detector, we propose a novel MIMO detector. This detector is based on a sort-free strategy, and utilizes a new modified real-valued decomposition ordering (M-RVD) scheme.

4.1.1 Tree Traversal for Flex-Sphere Detection

In order to address the sorting challenge, we propose using a sort-free detector. With this technique, the long sorting operation is effectively simplified to a minimum-finding operation. The detailed steps of this algorithm are described below:

Algorithm 1 Flex-Sphere Detection Algorithm

Input: $\mathbf{R}$, $\mathbf{y}'$
$T_{M+1}(\mathbf{s}^{(M+1)}) = 0$
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{L}' \leftarrow \emptyset$
$i \leftarrow M$
\\ Full expansion of the first level:
- Compute T_i with (4.4)
- $\mathcal{L} \leftarrow \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j = 1, ..., w'\}$
- $i \leftarrow i - 1$
\\ Full expansion of the second level:
- **for** each $(\mathbf{s}^{(i+1)}, T_{i+1}(\mathbf{s}^{(i+1)})) \in \mathcal{L}$, **repeat**
  - compute $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j$ children pairs, $j = 1, ..., w'$
  - $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j = 1, ..., w'\}$
- **end**
-$\mathcal{L} \leftarrow \mathcal{L}'$
-$\mathcal{L}' \leftarrow \emptyset$
\\ Minimum-based expansion of the next levels:
- **for** $i = M - 2$ down to $i = 1$, **repeat**
  - **for** each $(\mathbf{s}^{(i+1)}, T_{i+1}(\mathbf{s}^{(i+1)})) \in \mathcal{L}$, **repeat**
    - compute $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j$ children pairs, $j = 1, ..., w'$
    - $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min} \leftarrow \underset{\{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j=1,...,w'\}}{\text{argmin}} T_i(\mathbf{s}^{(i)})$
    - $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min}\}$
  - **end**
  - $\mathcal{L} \leftarrow \mathcal{L}'$
  - $\mathcal{L}' \leftarrow \emptyset$
  - $i \leftarrow i - 1$
- **end**

- $(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{detected} \leftarrow \underset{\mathcal{L}}{\text{argmin}} \; T_i(\mathbf{s}^{(i)})$

An example of this algorithm is illustrated in Figure 4.1 for a virtual 4×4, 64-QAM system. Note that as described above, the first two levels are fully expanded to guarantee high performance; whereas for the following levels, only the best candidate in the children list of a parent node is expanded. In other words, after passing the first two levels, w_{M_T} nodes are expanded, and for each of those w_{M_T} nodes, the best child node among its w'_M children nodes is selected as the surviving node. Therefore, the new node list would contain w_{M_T} nodes in the third level. These w_{M_T} nodes are expanded in a similar way to the fourth level, and this procedure continues until the very last level, where the minimum-distance node is taken as the detected node.

Moreover, from the Schnorr-Euchner (SE) ordering [25], we know that finding

$$(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_{min} \leftarrow \underset{\{(\mathbf{s}^{(i)}, T_i(\mathbf{s}^{(i)}))_j | j=1,\ldots,w'_i\}}{\operatorname{argmin}} T_i(\mathbf{s}^{(i)})$$

basically corresponds to finding the real-valued constellation point closest to $\frac{1}{R_{ii}} b_{i+1}(\mathbf{s}^{(i+1)})$; see Eq. (4.6). Thus, the long sorting of K-best is avoided.

4.1.2 Modified Real-Valued Decomposition Ordering

For the sort free detector described in the preceding section, we propose using a novel real-valued decomposition (M-RVD) ordering which improves the BER performance compared to the ordering given in Eq. (4.1). The new decomposition is summarized as:

$$\hat{\mathbf{y}} = \hat{\mathbf{H}}\hat{\mathbf{s}} + \hat{\mathbf{n}} \tag{4.7}$$

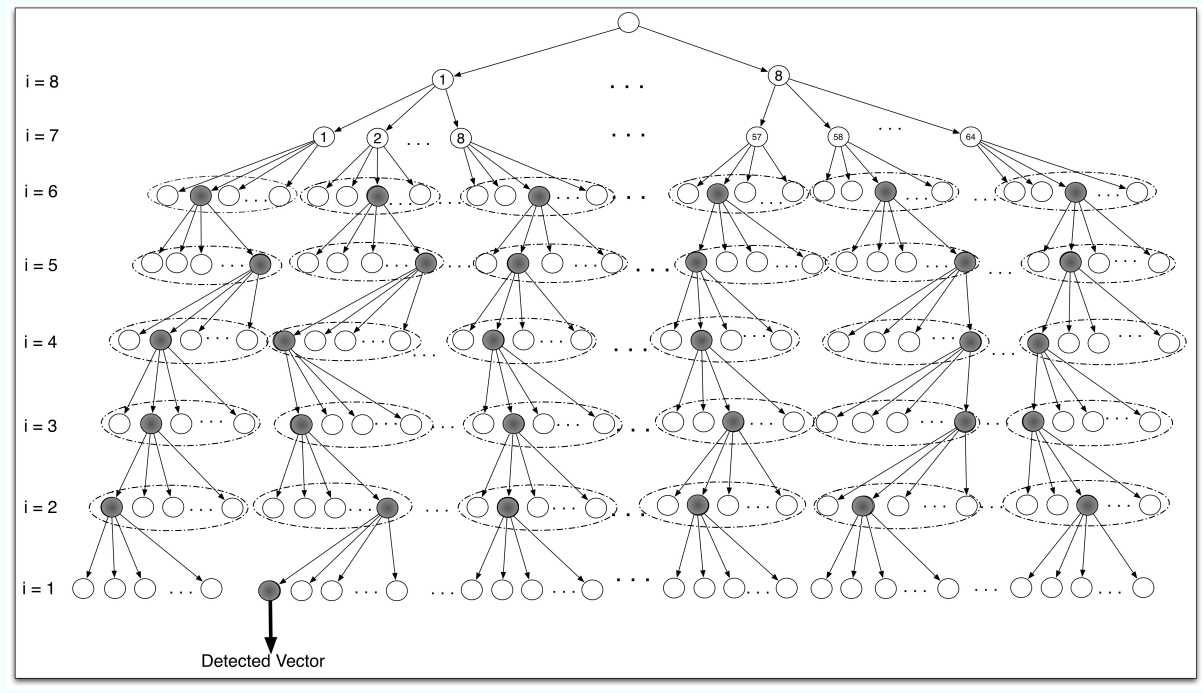

Figure 4.1 Flex-Sphere algorithm for a 64-QAM, 4×4 system. The topmost two levels are fully expanded. The nodes marked with black are the minimum in their own set, where each set is denoted by dashed line. Note that because of the real-valued decomposition, each node has only $\sqrt{64} = 8$ children. Also, the number of tree levels are $M = 2 \times M_T = 8$.

or,

$$\begin{pmatrix} \Re(\tilde{y}_1) \\ \Im(\tilde{y}_1) \\ \Re(\tilde{y}_2) \\ \Im(\tilde{y}_2) \\ . \\ . \\ . \\ \Re(\tilde{y}_{M_R}) \\ \Im(\tilde{y}_{M_R}) \end{pmatrix} = \hat{\mathbf{H}} \begin{pmatrix} \Re(\tilde{s}_1) \\ \Im(\tilde{s}_1) \\ \Re(\tilde{s}_2) \\ \Im(\tilde{s}_2) \\ . \\ . \\ . \\ \Re(\tilde{s}_{M_T}) \\ \Im(\tilde{s}_{M_T}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{n}_1) \\ \Im(\tilde{n}_1) \\ \Re(\tilde{n}_2) \\ \Im(\tilde{n}_2) \\ . \\ . \\ . \\ \Re(\tilde{n}_{M_R}) \\ \Im(\tilde{n}_{M_R}), \end{pmatrix} \tag{4.8}$$

where $\hat{\mathbf{H}}$ is the permuted channel matrix of Eq. (4.2) whose columns are reordered to match the other vectors of the new decomposition ordering in Eq. (4.7). It is worth noting that since the difference between RVD and M-RVD is the grouping of the signals, there is no extra computational cost associated with this novel ordering.

Note that with the modified real-valued decomposition (M-RVD) ordering, the first two levels correspond to the in-phase and quadrature parts of the same complex symbol whereas in the conventional real-valued decomposition scenario, the first two levels of the tree correspond to the quadrature parts of two different complex symbols. A careful look at the tree traversal scheme of the preceding section shows that since the first two levels of the tree are fully expanded, the error performance of the scheme

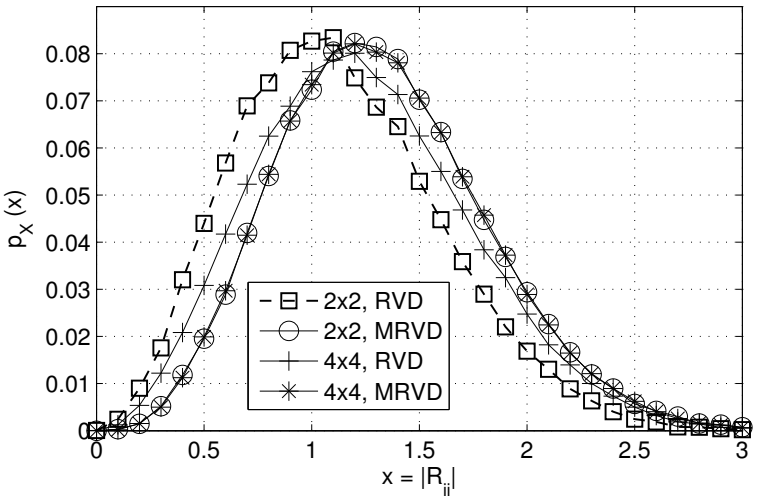

Figure 4.2 Probability density function of $R_{6,6}$ for 4×4 and $R_{2,2}$ for 2×2 when either conventional RVD or the proposed RVD are used. Note the shift of the curves when M-RVD is used.

heavily depends on the third level of the tree. Therefore, rather than using the magnitude of $R_{M,M}$ as a metric to choose the decomposition ordering scheme, which justifies the conventional real-valued decomposition (RVD) [7], we need to look at the behavior of the third lowest diagonal element of the $\mathbf{R}$ matrix. As demonstrated in Figure 4.2, there is an increase in the magnitude of $R_{M-2,M-2}$ when using M-RVD, hence M-RVD is a better choice than the conventional RVD. The impact of M-RVD on the BER performance is discussed in the next sections.

Table 4.1 Comparison of the latency and the operation counts between the conventional K-best and the proposed Flex-Sphere detector.

	K-best	Flex-Sphere detector
Compare-select	$f_K(Kw')(M-3)+f_K(w)+f_1(Kw')$	$w'f_1(w')(M-3)+f_1(w)$
Addition	$2w'+2w+2Kw'(M-2)+K(M(M-1)/2-1)$	$2w'+w+w.w'(M(M+1)/2-3)$
Multiplication	$w'+w+Kw'(M-2)$	-
Latency	$\sum_{m=0}^{K-1}\lceil\log(Kw'-m)\rceil$	$\lceil\log w'\rceil$
Example (16-QAM, $K=4$)	16	2
Example (16-QAM, $K=5$)	24	2

4.2 Complexity Comparison

In order to compare the complexity of the proposed MIMO detector described in the preceding section, versus the conventional K-best technique, we consider the number of operations, the relative latency reduction, and the architecture advantages of the proposed detector.

4.2.1 Number of Operations

In this section, we compute the number of operations required to complete the detection process. Since the channel matrix typically changes at a much slower rate than the received signal vector, we make the assumption that simple channel matrix operations, e.g., $R_{ij}s_j$ computations, are performed in a separate pre-processing unit. Note that this simply involves shift-add operations with $s_j \in \Omega'$. Also, as suggested in [8], we make the assumption that all the PED norms are approximated by ℓ^1-norms to avoid the squarers and multipliers. Therefore, the only major high rate detector

operations are compare-select for either sorting or minimum-findings, addition and multiplication.

Note that in order to achieve minimum latency, we make the assumption that both detectors use cascaded minimum-finders to sort a list. Therefore, in order to find the best K elements of a list of size l; K cascaded minimum finders are required. So, the number of operations required to sort the best K candidates of a list of size l, denoted by $f_K(l)$ in Table 4.1, is given by

$$f_K(l) = K \times l - \frac{K(K+1)}{2}. \tag{4.9}$$

Given the above assumptions, the total number of operations for the K-best scenario and the proposed Flex-Sphere scheme are given in Table 4.1.

1. Compare-Select: The K-best method requires finding the best K nodes among Kw' candidates in $(M-3)$ of the levels, i.e. $f_K(Kw')(M-3)$ operations; whereas, the Flex-Sphere only needs to compute the minimum nodes among w' nodes for w' groups in those $M-3$ levels, i.e. $w'f_1(w')(M-3)$ operations. Moreover, the best node is chosen among Kw' nodes, i.e. $f_1(Kw')$ operations, in the last level of the K-best tree, and among $w'^2 = w$ nodes, i.e. $f_1(w)$ operations, in the Flex-Sphere. While the second level requires finding the best K nodes among the w children, i.e. $f_K(w)$ operations in the K-best structure, the Flex-Sphere does not need such sorting since it is fully expanding that level.

2. Addition: For the K-best scenario, assuming that $K > w'$, which ensures higher performance, and based on Eq. (4.5), level $i = M$ requires w' addition operations, level $i = M - 1$ requires $w'(1 + w')$ operations, and the rest of the levels each need $K(M - i + w')$ addition operations. Moreover, based on Eq. (4.4), level $i = M - 1$ needs w addition operations, and each of the remaining levels, $i = M - 1, ..., 1$, needs Kw' operations. Therefore, the total number of adders needed for the K-best detection scheme is given in Table 4.1. A similar approach will yield the total number of additions required for the Flex-Sphere detection.

3. Multiplication: The Flex-Sphere uses ℓ^1-norms, and thus, does not need to use the FPGA multipliers; whereas, the K-best scheme needs to compute w' ℓ^2-norms in the first level, w norms in the second level, and Kw' norms in the remaining $(M - 2)$ levels.

In order to compute the final operation count, comparators are assumed to have unit complexity, and adders to have twice complexity as that of comparators. Multipliers are needed to implement the squarers, and for the wordlengths that we are interested in, i.e. 16 bits, they can be assumed to be ten times more complex than additions. It is worth noting that other relative complexity coefficients would yield similar general results. Based on these relative complexities, the number of operations are plotted for different numbers of antennas in Figure 4.3. The operation

count increases for higher K values because higher K means higher number of visited nodes per level; therefore, higher K requires larger computations. Note that except for small K values, the computation overhead of the conventional K-best scheme is considerably more than the proposed Flex-Sphere scheme. More details on the BER performance comparisons will be presented in section 5.5.4.

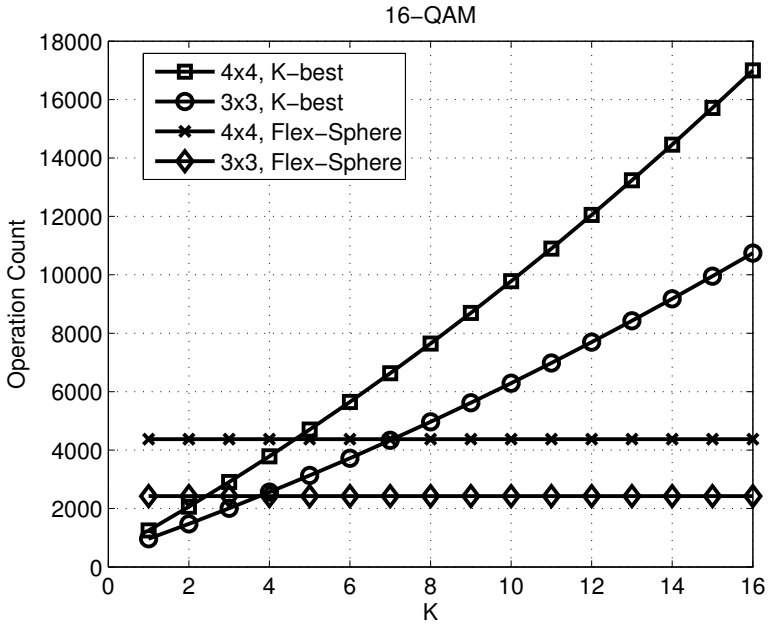

Figure 4.3 Comparison of the number of operations between the proposed scheme and K-best for different values of K and different number of antennas. The 16-QAM modulation is assumed.

4.2.2 Latency

High latency decreases the data rate in feedback based receivers. For instance, for iterative detector/decoder structures, where the detector uses the feedback data from the decoder to improve the detection performance, higher detection/decoding latency reduces the data rate significantly. A similar argument applies to the overall receiver throughput when the interaction between the physical layer and MAC layer takes more cycles due to the higher physical layer latency. We compare the latency overhead of our proposed detector versus the conventional K-best detector, and show that the Flex-Sphere technique introduces significant latency reduction.

Note that if the detectors are fully parallelized for enhancing data rates, the conventional K-best detector requires K successive minimum finders. The first minimum finder needs to find the minimum among Kw' candidates, and therefore, has a latency of $Kw'-1$. The second one needs to find the minimum among $Kw'-1$, and therefore has a latency of $Kw'-2$, and so on. The proposed Flex-Sphere detector, however, requires only one level of minimum finder as it only needs to find the minimum, i.e. sorting with $K=1$. Thus, if we assume full parallelism for both types of detectors, the latency of the sorter that connects one of the middle levels of the tree to the next level is given in Table 4.1.

Notice the significant latency reduction that the proposed Flex-Sphere detector promises for the sorting after each level. Also, note that Table 4.1 represents only the

latency of one level; thus, for a 4×4 system, there would be $M - 3 = 2M_T - 3 = 5$ of such sorters, see Table 4.1.

4.2.3 Architecture

As pointed out earlier, Flex-Sphere significantly reduces the complexity over the simple solution where multiple detectors are used for different modulation and/or number of antennas. In general, assuming that the Flex-Sphere supports M_T antennas and a modulation order of w, then, the area reduction due to using Flex-Sphere is given by

$$\frac{A_{SD}}{A_{FS}} = \frac{N_m(M_T - 1 + f(M_T^2 - 1))}{(2M_T - 1)f + 1} \cdot PED_{scaling} \tag{4.10}$$

where A_{SD} corresponds to the area of the multiple sphere detectors, and A_{FS} corresponds to the area of the flex-sphere solution. The number of modulation orders that the Flex-Sphere supports is denoted by N_m, and f corresponds to the folding factor, which is usually set to $f = 8$ as shown later in this chapter. Note that the implementation of PED in the Flex-Sphere requires slight overhead to support configurability; therefore, the parameter $PED_{scaling} < 1$ represents this change in the area of the PED from regular sphere detection to Flex-Sphere. Typical values of $PED_{scaling}$ based on FPGA synthesis is 0.9. Figure 4.4 shows the area reduction using Flex-Sphere for some of the typical cases of wireless communication.

The common K-best sorting requires a bubble-sort architecture [66]. In this archi-

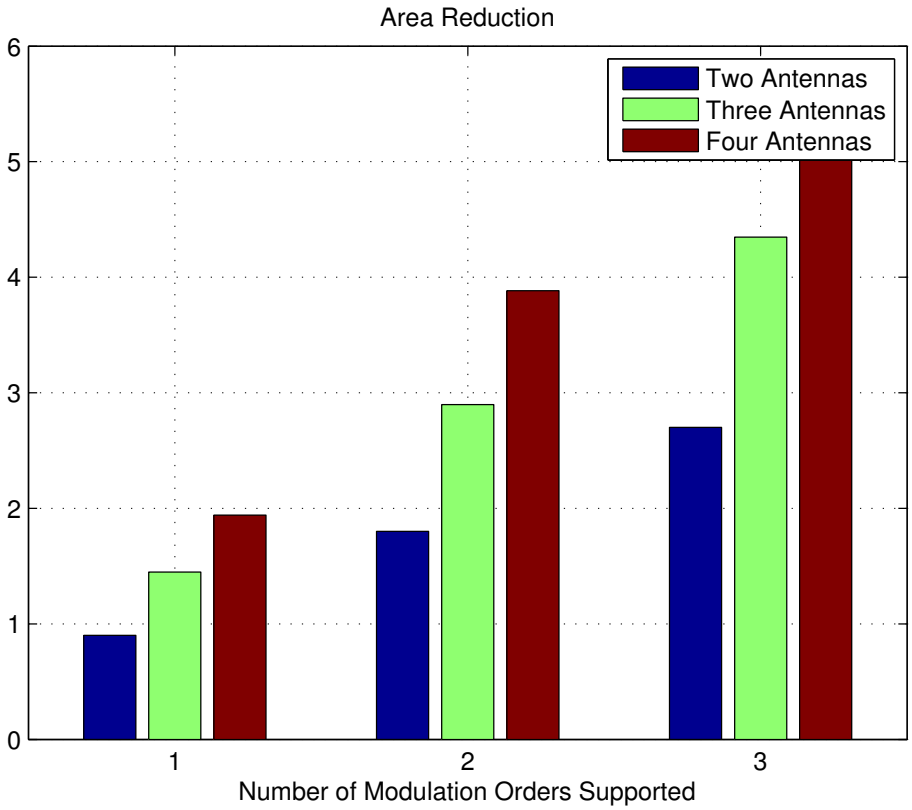

Figure 4.4 The area reduction using Flex-Sphere for $2, 3$ and 4 antennas. The vertical axis shows $\frac{A_{SD}}{A_{FS}}$ from Eq.(4.10).

tecture, all the nodes need to be passed into the sorter sequentially, and the process of the next level of the tree can not start until all the $K \times w'$ nodes are passed through the sequential sorter. Even semi-parallel sorters still require large area and cycles to finish the detection process, see Table 4.1 and Figure 4.3. With the Flex-Sphere technique, all the long size sortings are avoided. Moreover, the Flex-Sphere technique is amenable to parallelizing with less overhead than the K-best technique.

4.2.4 Simulation Results

For the BER simulations, the Rayleigh fading channel model is assumed, and the channel matrix is independent for each new transmission. The BER results of 4×4 and 3×3 systems are compared for a 16-QAM modulation scheme. Note that in order to conduct a fair performance comparison, the K values are chosen such that the K-best technique has similar number of operations as that of the proposed Flex-Sphere scheme, see Figure 4.3. Therefore, based on the results shown in Figure 4.3 and Table 4.1, K is set to 5 and 4 for the 4×4 and 3×3 systems, respectively.

The BER simulation results of Figure 4.5 suggest that the proposed Flex-Sphere scheme can improve the BER performance more than 5 dB compared to the conventional K-best technique in higher SNR regimes. Note that it was shown in the preceding sections that for a 4×4 case, the $K = 5$ scheme requires similar computational complexity as that of the Flex-Sphere scheme, and it requires 12 times more latency for sorting in each level compared to the proposed sort-free scheme. A similar

argument holds for a 3×3 system when $K = 4$. It is also worth noting that in both cases, the M-RVD ordering plays an important role in improving the performance.

4.3 FPGA Design of the Configurable Detector for SDR Handsets

In this section, the main features of the architecture and the FPGA implementation of the SDR handset detector are presented. We use Xilinx System Generator [6] to implement the proposed architecture. In order to support all the different number of antenna/user and modulation orders, the detector is designed for the maximal case, i.e. $M_T \times M_R$, 64-QAM case, and configurability elements are introduced in the design to support different configurations.

4.3.1 PED Computations

Computing the norms in (4.6) is performed in the PED blocks. Depending on the level of the tree, three different PED blocks are used: The PED in the first real-valued level, PED_1, corresponds to the root node in the tree, $i = M = 2M_T = 8$. The second level consists of $\sqrt{64} = 8$ parallel PED_2 blocks, which compute 8 PEDs for each of the 8 PEDs generated by PED_1; thus, generating 64 PEDs for the $i = 7$ level. Followed by this level, there are 8 parallel general PED computation blocks, PED_g, which compute the closest-node PED for all 8 outputs of each of the PED_2s. The next levels will also use PED_g. At the end, the Min_Finder unit detects the signal by finding the minimum of the 64 distances of the appropriate level. The block diagram

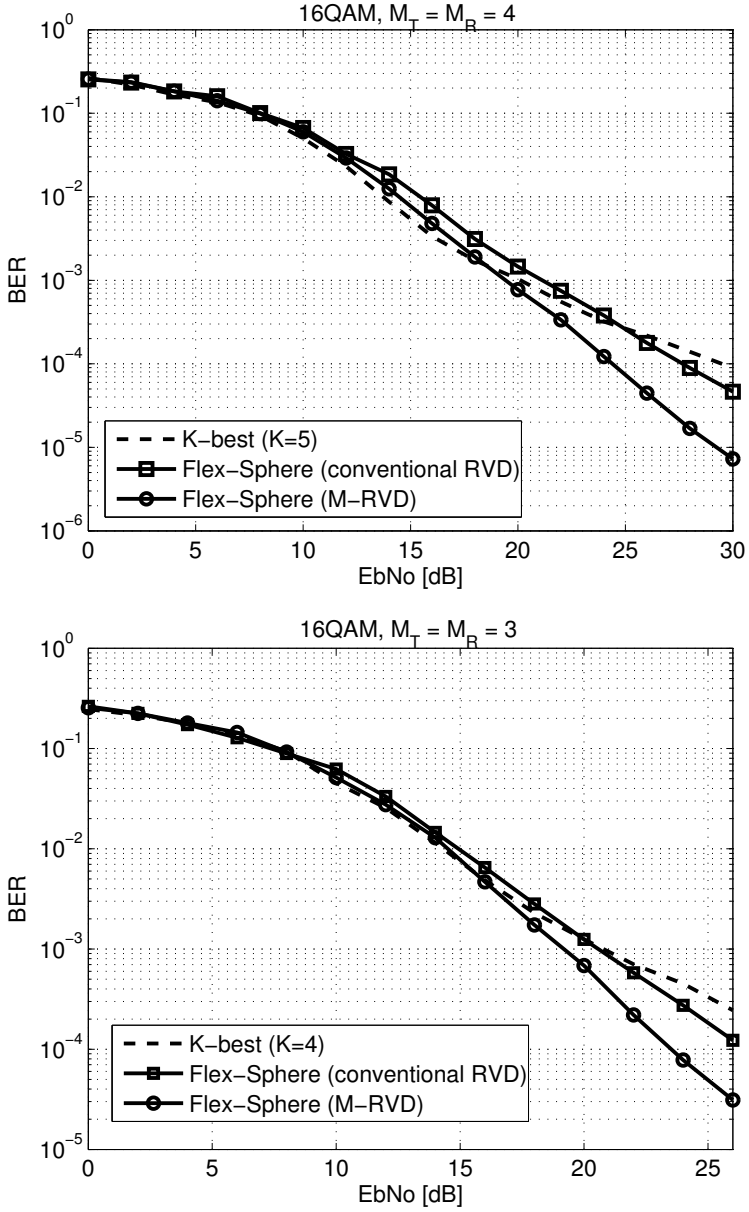

Figure 4.5 BER performance of the proposed detector with and without the novel ordering (M-RVD) described in section 4.1.2 assuming a 16-QAM modulation for both $M_T = M_R = 4$ and $M_T = M_R = 3$. The K-best implementation for $K = 5$ and $K = 4$ has similar computational complexity as that of the sort-free schemes for $M_T = 4$ and $M_T = 3$, respectively.

of this design is shown in Figure 4.6.

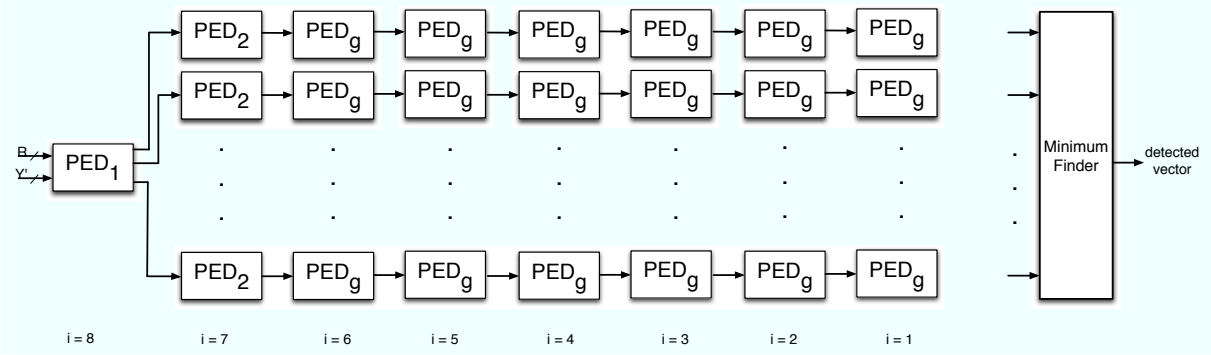

Figure 4.6 The block diagram of the Flex-Sphere. Note that there are M parallel PEDs at each level. The inputs to the Min_Finder is fed from the appropriate PED block, as described in section 4.3.2.

4.3.2 Configurable Design

In order to ensure the configurability of the Flex-Sphere, it needs to support different M_T as well as different modulation orders for different users. The configurability of the detector is achieved through two input signals, M_T and $q^{(i)}$, which control the number of antennas and the modulation order, respectively. These two inputs can change based on the system parameters at any time during the detection procedure. Therefore, this configurability is a real-time operation.

Number of Antennas

The M_T determines the number of detection levels, and it is set through the M_T input to the detector, which, in turn, would configure the Min_Finder appropriately. Therefore, the minimum finder can operate on the outputs of the corresponding level, and generate the minimum result. In other words, the multiplexers in each input of the Min_Finder block choose which one of the four streams of data should be fed into the Min_Finder. Therefore, the inputs to the Min_Finder would be coming from the $i = 5, 3$ or 1, if M_T is $2, 3$ or 4, respectively, see Figure 4.6.

The M_T input can change on-the-fly; thus, the design can shift from one mode to another mode based on the number of streams it is attempting to detect at any time. Moreover, as will be shown later, the configurability of the minimum finder guarantees that less latency is required for detecting smaller number of streams.

Modulation Order

In order to support different modulation orders per data stream, Flex-Sphere uses another input control signal $q^{(i)}$ to determine the maximum real value of the modulation order of the i-th level. Thus, $q^{(i)} \in \{1, 3, 7\}$. Moreover, since the modulation order of each level is changing, a simple comparison-thresholding can not be used to find the closest candidate for Schnorr-Euchner [25] ordering. Therefore, the following

conversion is used to find the closest SE candidate:

$$\tilde{s} = g(2[\frac{b+1}{2}] - 1) \tag{4.11}$$

where [.] represents rounding to the nearest integer, $b = (1/R_{ii}) \cdot b_{i+1}$ of Eq. (4.6), and $g(.)$ is

$$g(x) = \begin{cases} -q^{(i)}, & x \leq -q^{(i)}; \\ x, & -q^{(i)} \leq x \leq q^{(i)}; \\ q^{(i)}, & x \geq q^{(i)}. \end{cases} \tag{4.12}$$

All of these functions can be readily implemented using the available building blocks of the Xilinx System Generator, see Figure 4.7. Note that the multiplications/divisions are simple one-bit shifts.

For the first two levels, which correspond to the in-phase and quadrature components of the last antenna, the PED of the out-of-range candidates are simply overwritten with the maximum value; thus, they will be automatically discarded during the minimum-finding procedure.

4.3.3 Modified Real Valued Decomposition

Using the real-valued decomposition, the two extra adders that are required per each complex multiplication can be avoided, thus, avoiding the unnecessary FPGA slices on the addition operations. Moreover, while using the complex-valued operations require the SE ordering of [8], which would be a demanding task given the

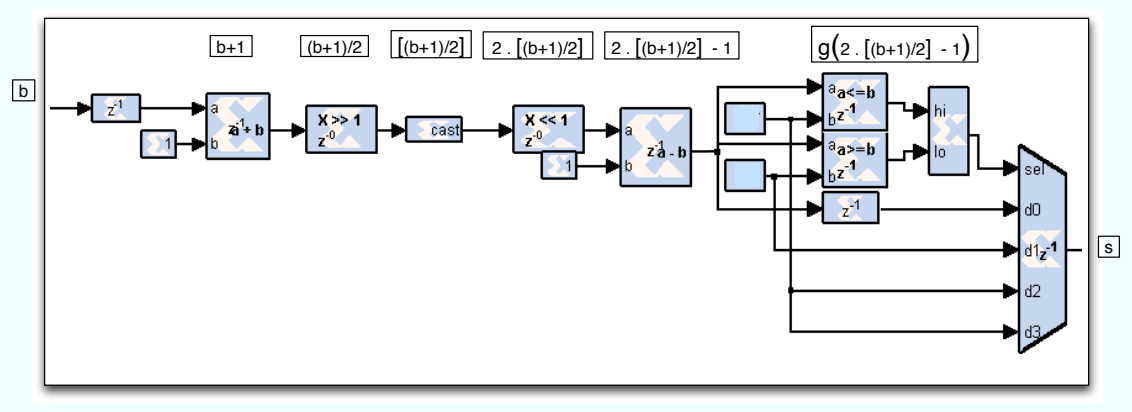

Figure 4.7 The pipelined System Generator block diagram for Eq. (4.11) in the PED$_g$ to support different modulation orders.

configurable nature of the detector; with the real-valued decomposition, the SE ordering can be implemented more efficiently and simply for the proposed configurable architecture as described earlier. Also, note that even though some of the multiplications can be replaced with shift-adds in an area-optimized ASIC design, as discussed in section 4.2; for an FPGA implementation, the appropriate design choice is to use the available embedded multipliers, commonly known as XtremeDSP and DSP48E in Virtex-4 and Virtex-5 devices.

It is noteworthy that if the conventional real-valued decomposition of (4.2) were employed, then, the results for a 2×2 system would have been ready only after going through all the in-phase tree levels and the first two quadrature levels. However, with

the modified real-valued decomposition (M-RVD), every antennas is isolated from other antennas in two consecutive levels of the tree. Therefore, there is no need to go through the latency of the unnecessary levels. Thus, using the M-RVD technique offers a latency reduction compared to the conventional real-valued decomposition.

4.3.4 Timing Analysis

Each of the PED_g blocks are responsible for expanding 8 nodes; thus, the folding factor of the design is $F = 8$. In order to ensure a high maximum clock frequency, several pipelining levels are introduced inside each of the PED computation blocks. The latency of the PED_1, PED_2 and PED_g blocks are $7, 17$ and 22, respectively. Note that the larger latency of the PED_g blocks is due to more multiplications required to compute the PEDs of the later levels. The Min_Finder block has a latency of 8.

As mentioned earlier, different values of M_T require different numbers of tree levels, which incur different latencies. The latencies of the three different configurations of M_T are presented in Table 4.2. In computing the latencies, an initial 8 cycles are required to fill up the pipeline path.

4.3.5 Implementation Results on WARP

The Wireless Open-access Research Platform (WARP) [4], which is a scalable and extensible programmable wireless platform, is a suitable platform for prototyping the detection algorithms. Each board can support up to four antennas, and if the boards

Table 4.2 Latency for different values of M_T.

M_T	Latency
$M_T = 2$	$8 + \text{PED}_1 + \text{PED}_2 + 2 \cdot \text{PED}_g + \text{Min\_Finder} = 84$
$M_T = 3$	$8 + \text{PED}_1 + \text{PED}_2 + 4 \cdot \text{PED}_g + \text{Min\_Finder} = 128$
$M_T = 4$	$8 + \text{PED}_1 + \text{PED}_2 + 6 \cdot \text{PED}_g + \text{Min\_Finder} = 172$

are stacked together to form a bigger node, they can support even more antennas. Several architecture-friendly wireless algorithms for handsets have been implemented and verified on this testbed, see Figure 4.8. The new version of this board is based on Virtex-4 FPGA, and Table 4.3 presents the System Generator implementation results of the Flex-Sphere on a Xilinx Virtex-4 FPGA, xc4vfx100-10ff1517 [6] for 16-bit precision. The maximum number of detectable streams is set to $M_T = 3$. The maximum achievable clock frequency is 250 MHz. Since the design folding factor is set to $F = 8$, the maximum achievable data rate, i.e. $M_T = 3$ and $w_i = 64$, is

$$D = \frac{M_T \cdot \log w}{F} \cdot f_{max} = 562.5 \text{ [Mbps]}. \tag{4.13}$$

4.3.6 Implementation Results for $M_T = 4$

Table 4.4 presents the System Generator implementation results of the Flex-Sphere on a Xilinx Virtex-5 FPGA, xc5vsx95t-3ff1136 [6] for 16-bit precision and

Table 4.3 FPGA resource utilization summary of the proposed Flex-Sphere for the Xilinx Virtex-4, xc4vfx100-10ff1517, device.

No. of Antennas	2, 3
Modulation Order	$\{4, 16, 64\}$-QAM
Max. Data Rate	562.5 Mbps
Number of Slices	18,825/42,176 (44%)
Number of Slice FFs	23,961/84,352 (28%)
Number of LUTs	30,297/84,352 (35%)
Number of DSP48E	129/160 (80%)
Max. Freq.	250 MHz

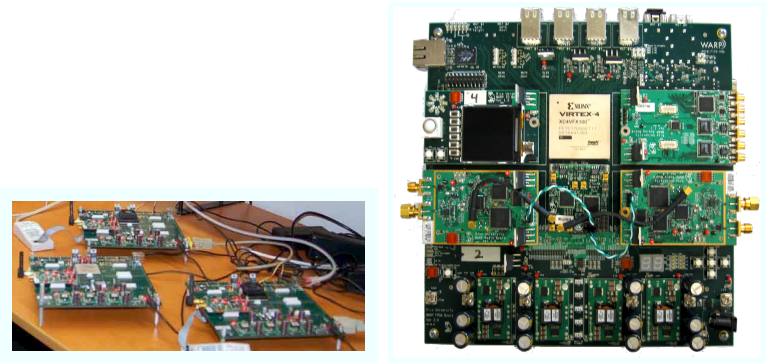

Figure 4.8 A WARP board with four daughtercard slots, and the boards used in a cooperative setup on the left.

$M_T = 4$. The maximum achievable clock frequency is 285.71 MHz. Since the design folding factor is set to $F = 8$, the maximum achievable data rate, i.e. $M_T = 4$ and $w_i = 64$, is

$$D = \frac{M_T \cdot \log w}{F} \cdot f_{max} = 857.1 \text{ [Mbps]}. \tag{4.14}$$

This table also presents the implementation results of a previously reported 64-QAM, 4×4 system [68]. While the proposed Flex-Sphere is implemented on a different FPGA device, due to its relatively larger size, it can support different numbers of antennas and modulation orders, and achieves the high data rate requirements of various wireless standards.

Table 4.5 summarizes the data rates for all of the different scenarios of the $M_T = 4$,

Virtex-5, implementation.

Table 4.4 Comparison of the system support and FPGA resource utilization of the proposed Flex-Sphere vs. optimized FSD-B [68].

Design	Flex-Sphere	Optimized FSD-B [68]
Device	XC5VSX95	XC2VP70
No. of Antennas	2, 3, 4	4
Modulation Order	$\{4, 16, 64\}$-QAM	64-QAM
Max. Data Rate	857.1 Mbps	450 Mbps
BER = 10^{-4} @ SNR =	= 25 dB	= 25 dB
Number of Slices	11,604/14,720 (78 %)	24,815/33,088 (74 %)
Number of Registers/FFs	27,115/58,880 (46 %)	39,800/66,176 (60 %)
Number of Slice LUTs	33,427/58,880 (56 %)	31,759/66,176 (47 %)
Number of DSP48E/Multipliers	321/640 (50 %)	252/328 (88 %)
Number of block RAMs	0 (0 %)	88/328 (26 %)
Max. Freq.	285.71 MHz	150 MHz

4.4 Simulation Results

In this section, we present the simulation results for the Flex-Sphere, and compare the performance of the FPGA fixed-point implementation with that of the optimum

Table 4.5 Data rate for different configurations of the 4×4, Table 4.4, implementation.

	4-QAM	16-QAM	64-QAM
$M_T = 2$	142.7 Mbps	285.7 Mbps	428.4 Mbps
$M_T = 3$	214.1 Mbps	428.4 Mbps	642.7 Mbps
$M_T = 4$	285.7 Mbps	571.4 Mbps	857.1 Mbps

floating-point maximum-likelihood (ML) results. Prior to the M-RVD, introduced in section 4.1, we employ the channel ordering of [67] to further close the gap to ML. Also, we make the assumption that all the streams are using the same modulation scheme. We assume a Rayleigh fading channel model, i.e. complex-valued channel matrices with the real and imaginary parts of each element drawn from the normal distribution.

In order to ensure that all the antennas in the receiver have similar average received SNR, and none of the user's messages are suppressed with other messages, a power control scheme is employed. Figure 4.9 shows the simulation results for the maximal 4×4 configuration. As can be seen, the proposed hardware architecture implementation performs within, at most, 1 dB of the optimum maximum-likelihood detection. Moreover, the flexibility provided by Flex-Sphere allows performing coop-

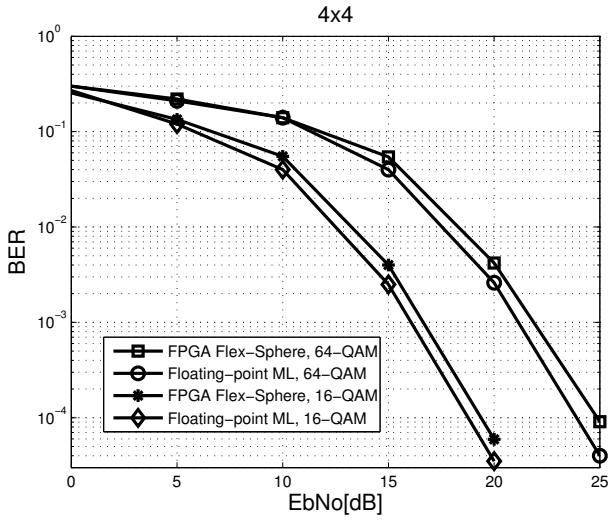

Figure 4.9 BER plots comparing the performance of the floating-point maximum likelihood (ML) with the the FPGA implementation. Note that the channel pre-processing of [67] is employed to improve the performance.

erative partial detection in relay networks.

Chapter 5
Cooperative Partial Detection Using MIMO Relays

Throughout this chapter, we use the same assumptions of three node networks, as outlined in the earlier chapters. In the first time slot, the source broadcasts its message to both the relay and the destination; and in the second time slot, the relay, using a $ef \leq M_r$ subset of its antennas, transmits its message to the destination while the source is silent. The *expansion factor*, ef, corresponds to the number of *utilized* antennas in the relay during the second time slot. The choice of ef and its impact on the performance and complexity will be discussed in detail in the next sections. The transmitted vector from the source is of length M_s and the source uses a spatial multiplexing scheme to transmit different streams, i.e. modulation symbols, on different antennas.

5.1 Conventional Full Detect-and-Forward with MIMO Relays

In this section, we present the symbol-level detector in the relay and destination [62, 61]. In the full detect-and-forward (FDF), the source transmits $\mathbf{x}_s$ in the first time slot, and the relay and destination receive their copies of the transmitted vector, $\mathbf{y}_r$ and $\mathbf{y}_d^{(1)}$. Then, the relay performs full sphere detection, as described in the previous section, on its received vector, $\mathbf{y}_r$, to find $\tilde{\mathbf{x}}_s$, where $\tilde{\mathbf{x}}_s$ is equal of $\mathbf{x}_s$ in an error-free

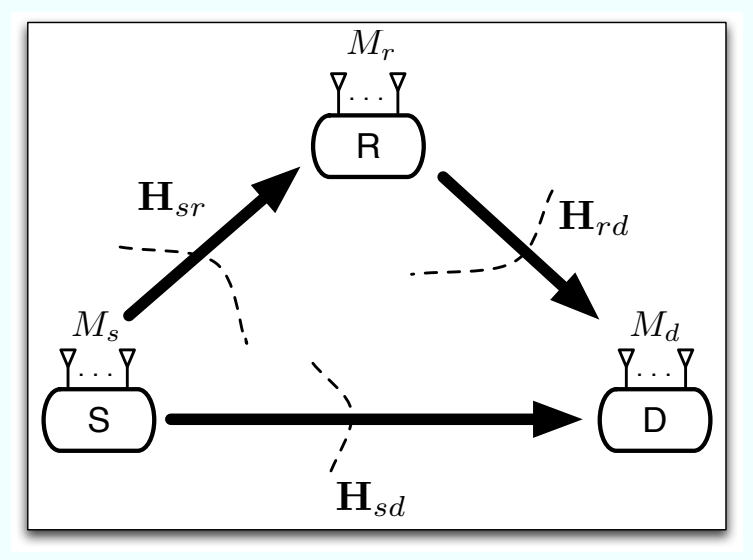

Figure 5.1 A relay network with three nodes: source, relay and destination. The respective channel matrices are denoted by $\mathbf{H}_{sr}$, $\mathbf{H}_{rd}$ and $\mathbf{H}_{sd}$.

detection:

$$\tilde{\mathbf{x}}_s = \underset{\mathbf{b}\in\mathcal{O}^m}{\operatorname{argmin}} \left|\left| \mathbf{y}_r - \mathbf{H}_{sr}\mathbf{b} \right|\right|^2. \tag{5.1}$$

The norm in (5.1) can be re-written as [34]

$$\begin{aligned} D(\mathbf{b}) &= \| \mathbf{y}_r - \mathbf{H}_{sr}\mathbf{b} \|^2 \\ &= \| \mathbf{Q}^H\mathbf{y}_r - \mathbf{R}\mathbf{b} \|^2 = \sum_{i=M_s}^{1} \left| y_i{}' - \sum_{j=i}^{M_s} R_{i,j} b_j \right|^2 \end{aligned} \tag{5.2}$$

where $\mathbf{H}_{sr} = \mathbf{QR}$, $\mathbf{QQ}^H = \mathbf{I}$ and $\mathbf{y}' = \mathbf{Q}^H\mathbf{y}_r$. Throughout this chapter, we will use the superscript ^{H} to denote the matrix Hermitian transpose. This minimization process can be performed in a depth-first tree search [34].

Finally, the relay transmits the $\mathbf{x}_r = \tilde{\mathbf{x}}_s$ in the second time slot to the destination, using the same modulation order. The received vector at the destination from the relay is denoted by $\mathbf{y}_d^{(2)}$. The destination can now combine the received copies from the source and relay and perform a sphere detection on the newly formed combined vector.

We will now derive the combination procedure. Given the two received copies in the destination, the Maximum-Likelihood detector is

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmax}} P\{\mathbf{x}|\mathbf{y}_d^{(2)}, \mathbf{y}_d^{(1)}, \mathbf{H}_{rd}, \mathbf{H}_{sd}, \mathbf{x}_s = \tilde{\mathbf{x}}_s\} \tag{5.3}$$

which, given the equal noise power in different links, is equivalent to

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}} (\| \mathbf{y}_d^{(2)} - \mathbf{H}_{rd}\mathbf{x} \|_2^2 + \| \mathbf{y}_d^{(1)} - \mathbf{H}_{sd}\mathbf{x} \|_2^2). \tag{5.4}$$

After expanding each of the norms in (5.4) and regrouping the different terms, Eq. (5.4) can be rewritten as

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}}(\| \mathbf{y}_{FDF} - \mathbf{H}_{FDF}\mathbf{x} \|_2^2) \tag{5.5}$$

where the equivalent channel matrix, $\mathbf{H}_{FDF}$, and the equivalent received vector, $\mathbf{y}_{FDF}$, are given by

$$\mathbf{H}_{FDF} = (\mathbf{H}_{sd}^H\mathbf{H}_{sd} + \mathbf{H}_{rd}^H\mathbf{H}_{rd})^{1/2} \tag{5.6}$$

$$\mathbf{y}_{FDF} = \mathbf{H}_{FDF}^{-1}(\mathbf{H}_{sd}^H\mathbf{y}_d^{(1)} + \mathbf{H}_{rd}^H\mathbf{y}_d^{(2)}). \tag{5.7}$$

It is worth noting that Eq. (5.6) and (5.7) are essentially similar to performing a MIMO Maximal-Ratio Combining (MRC), followed by whitening the colored noise [35]. The equivalent received vector and channel matrix can also be computed by concatenating the received signals and channel matrices:

$$\mathbf{H}_{FDF} = \begin{bmatrix} \mathbf{H}_{sd} \\ \\ \mathbf{H}_{rd} \end{bmatrix}, \tag{5.8}$$

$$\mathbf{y}_{FDF} = \begin{bmatrix} \mathbf{y}_d^{(1)} \\ \\ \mathbf{y}_d^{(2)} \end{bmatrix}. \tag{5.9}$$

While the concatenation process of Eq. (5.8)-(5.9) does not require the per-vector combining of Eq. (5.6)-(5.7), it increases the size of the effective channel matrix, and

thus, requires more resources for the QR decomposition. However, since the QR decomposition needs to happen at the channel updating rate, as opposed to symbol vector rate, it generally leads to a less complex procedure.

The soft values can then be computed according to [21]:

$$\begin{aligned} L_E(u_k|\mathbf{y}_{FDF}) \approx & \frac{1}{2} \max_{\mathbf{u}\in\mathcal{L}\cap\mathbf{U}_{\mathbf{k},+\mathbf{1}}} \left\{ -\frac{1}{\sigma^2}||\mathbf{y}_{FDF} - \mathbf{H}_{FDF}\mathbf{x}||^2 \right\} \\ & -\frac{1}{2} \max_{\mathbf{u}\in\mathcal{L}\cap\mathbf{U}_{\mathbf{k},-\mathbf{1}}} \left\{ -\frac{1}{\sigma^2}||\mathbf{y}_{FDF} - \mathbf{H}_{FDF}\mathbf{x}||^2 \right\}, \end{aligned} \tag{5.10}$$

where $\mathcal{L}$ is the list of possible vectors, σ^2 is the noise variance, $\mathbf{U}_{\mathbf{k},+\mathbf{1}}$ is the set of $2^{m\log w-1}$ bits of vector $\mathbf{u}$ with $u_k = +1$, while $\mathbf{U}_{\mathbf{k},-\mathbf{1}}$ is similarly defined. Note that the performance of such a detector and decoder pair will be further improved if the detector and decoder, iteratively, pass the LLR information between each other [21]. However, since the focus of this chapter is on the cooperative aspect of the detection process, and in order not to complicate the parameters, we choose a no-iteration case.

Figure 5.2 summarizes the steps of the full detect-and-forward.

5.2 Reducing Complexity Using Cooperative Partial Detection with MIMO Relays

In this section, we propose cooperative partial detection (CPD) as a low-complexity strategy for relays with limited resources. Cooperative partial detection (CPD) is based on partial sphere detection in the relay to facilitate the cooperative detection strategy.

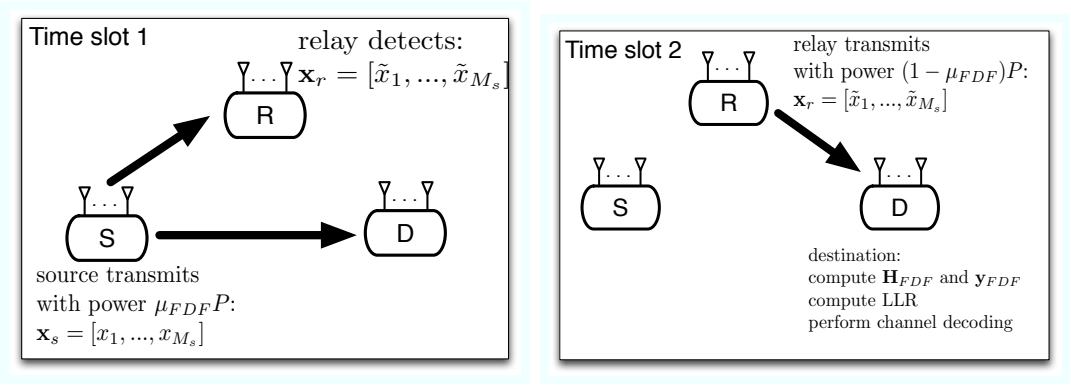

Figure 5.2 Full Detect-and-Forward (FDF) through MIMO relay node. In the first time slot, the relay receives a copy of the source multi-stream data, and detects it, and forwards the detected data. In the second time slot, the receiver combines the multiple copies as described earlier to compute the LLR values. We denote the power splitting ratio by μ_{FDF}.

5.2.1 Partial Sphere Detection in the Relay

In order to reduce the relay overhead, we propose partial sphere detection (P-SD), where the relay visits only a subset of the tree levels, as opposed to all the levels. Our proposed partial sphere detection (P-SD) requires similar pre-processing operations as that of the conventional sphere detector: the QR decomposition triangularizes the channel matrix, and the tree traversal starts from the top level, $i = M_s$, where M_s is the number of transmit antennas. Unlike the conventional sphere detection method, the tree traversal of the partial sphere detection method terminates in one of the middle levels, and the corresponding minimum distance at that level is considered as the *partial* detected symbol vector. We call the number of visited antennas the *expansion factor*, *ef*, and use *ef* antennas of the relay to transmit those messages. Figure 5.3 shows this process for an example case with 16-QAM modulation, and expansion factor of 2.

In other words, instead of transmitting $\mathbf{x}_r = \tilde{\mathbf{x}}_s$, as in FDF, the relay now transmits only *ef* symbols, $\mathbf{x}_r = [\tilde{x}_1, ..., \tilde{x}_{ef}]^T$, where the superscript ^{T} denotes the vector transpose operation.

In order to understand the computational savings of the P-SD, we should note that the complexity of sphere detection, in terms of computation count, can be modeled

as:

$$C_{SD} = \sum_{i=M_s}^{1} C_i E\{D_i\}, \tag{5.11}$$

where C_i corresponds to the computation count for one node in level i, and $E\{D_i\}$ is the average number of visited nodes in level i. Based on Eq. (5.2), it is clear that C_i is larger for the nodes closer to the bottom of the tree, i.e. $C_{i+1} < C_i$. Therefore, P-SD reduces the total complexity in the relay by not only reducing the total number of visited nodes, but also by limiting the search to the nodes located at the top of the tree with less computation per node.

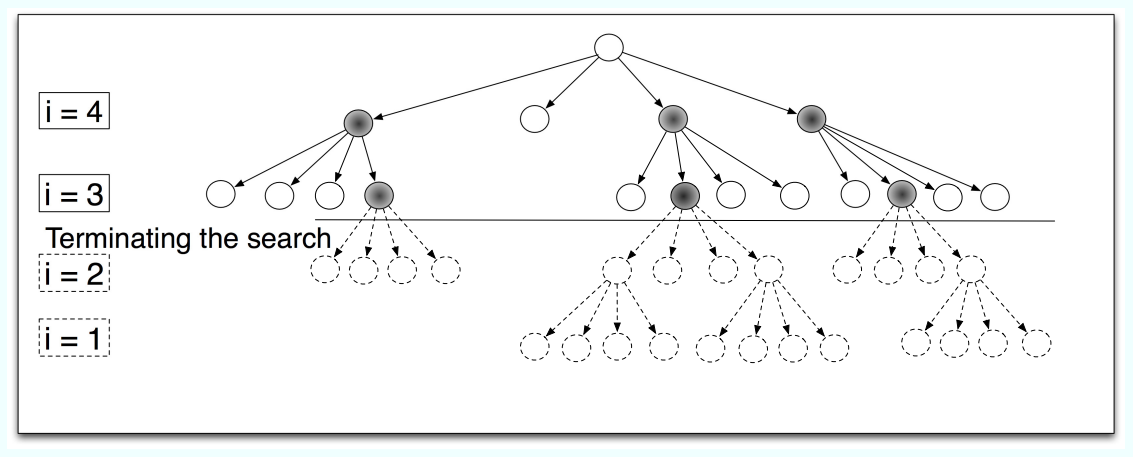

Figure 5.3 The tree structure for a partial sphere detector with the expansion factor of two, $ef = 2$. Each node has 16 children for the example case of 16-QAM modulation.

5.2.2 Cooperative Partial Detection in the Destination

In the symbol combining method, the destination combines the two received vector, $\mathbf{y}_d^{(1)}$ and $\mathbf{y}_d^{(2)}$, as shown below.

We first break the original transmitted vector into two parts:

$$\mathbf{x}_s = \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \\ \mathbf{x}_2 \end{bmatrix}, \tag{5.12}$$

where

$$\mathbf{x}_1 = \begin{bmatrix} x_1 \\ \vdots \\ x_{ef} \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} x_{ef+1} \\ \vdots \\ x_{M_s} \end{bmatrix}, \tag{5.13}$$

and denote the relay's transmitted vector as

$$\mathbf{x}_r = \tilde{\mathbf{x}}_1 = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_{ef} \end{bmatrix}. \tag{5.14}$$

We also split the source-destination channel matrix into two parts according to (5.12):

$$\mathbf{H}_{sd} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix}. \tag{5.15}$$

Similar to Eq. (5.4), assuming perfect detection in the relay, i.e. $\tilde{\mathbf{x}}_1 = \mathbf{x}_1$, the symbol level maximum-likelihood solution can be written as the following minimization

problem:

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}}(\parallel \mathbf{y}_d^{(2)} - \mathbf{H}_{rd}\mathbf{x}_1 \parallel_2^2 + \parallel \mathbf{y}_d^{(1)} - \mathbf{H}_1\mathbf{x}_1 - \mathbf{H}_2\mathbf{x}_2 \parallel_2^2). \tag{5.16}$$

Starting from Eq. (5.16), we can expand the norms, and keep the terms that depend on $\mathbf{x}$:

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}} D(\mathbf{x}) = \tag{5.17}$$

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}}(\parallel \mathbf{y}_d^{(2)} - \mathbf{H}_{rd}\mathbf{x}_1 \parallel_2^2 + \parallel \mathbf{y}_d^{(1)} - \mathbf{H}_1\mathbf{x}_1 - \mathbf{H}_2\mathbf{x}_2 \parallel_2^2) =$$

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}}(A - B - B^H + g(\mathbf{y}_d^{(1)}, \mathbf{y}_d^{(2)})), \tag{5.18}$$

where $g(.)$ contains those terms that do not depend on $\mathbf{x}$ and, hence, will not affect the solution, and A and B are given by:

$$\begin{aligned} A &= \mathbf{x}_2^H\mathbf{H}_2^H\mathbf{H}_2\mathbf{x}_2 + \mathbf{x}_1^H\mathbf{H}_{rd}^H\mathbf{H}_{rd}\mathbf{x}_1 + \mathbf{x}_1^H\mathbf{H}_1^H\mathbf{H}_1\mathbf{x}_1 \\ &+ \mathbf{x}_1^H\mathbf{H}_1^H\mathbf{H}_2\mathbf{x}_2 + \mathbf{x}_2^H\mathbf{H}_2^H\mathbf{H}_1\mathbf{x}_1 \end{aligned} \tag{5.19}$$

$$\begin{aligned} B &= \mathbf{x}_2^H\mathbf{H}_2^H\mathbf{y}_d^{(1)} + \mathbf{x}_1^H\mathbf{H}_1^H\mathbf{y}_d^{(1)} + \mathbf{x}_1^H\mathbf{H}_{rd}^H\mathbf{y}_d^{(2)} \\ &= \mathbf{x}_1^H(\mathbf{H}_1^H\mathbf{y}_d^{(1)} + \mathbf{H}_{rd}^H\mathbf{y}_d^{(2)}) + \mathbf{x}_2^H\mathbf{H}_2^H\mathbf{y}_d^{(1)}. \end{aligned} \tag{5.20}$$

Comparing Eq. (5.18) with

$$\parallel \mathbf{y}_{CPD} - \mathbf{H}_{CPD}\mathbf{x} \parallel_2^2 =$$

$$\parallel \mathbf{y}_{CPD} \parallel_2^2 - \mathbf{x}^H\mathbf{H}_{CPD}^H\mathbf{y}_{CPD} - \mathbf{y}_{CPD}^H\mathbf{H}_{CPD}\mathbf{x} + \mathbf{x}^H\mathbf{H}_{CPD}^H\mathbf{H}_{CPD}\mathbf{x} \tag{5.21}$$

shows that the original problem in (5.17) is equivalent to

$$\underset{\mathbf{x}\in\mathcal{O}^{M_s}}{\operatorname{argmin}}(\parallel \mathbf{y}_{CPD} - \mathbf{H}_{CPD}\mathbf{x} \parallel_2^2) \tag{5.22}$$

if we set

$$\mathbf{H}_{CPD} = \begin{bmatrix} \mathbf{H}_1^H\mathbf{H}_1 + \mathbf{H}_{rd}^H\mathbf{H}_{rd} & \mathbf{H}_1^H\mathbf{H}_2 \\ \mathbf{H}_2^H\mathbf{H}_1 & \mathbf{H}_2^H\mathbf{H}_2 \end{bmatrix}^{1/2} \tag{5.23}$$

$$\mathbf{y}_{CPD} = \mathbf{H}_{CPD}^{-1} \begin{bmatrix} \mathbf{H}_1^H\mathbf{y}_d^{(1)} + \mathbf{H}_{rd}^H\mathbf{y}_d^{(2)} \\ \mathbf{H}_2^H\mathbf{y}_d^{(1)} \end{bmatrix}. \tag{5.24}$$

Similar to Eq. (5.8)-(5.9), the equivalent channel matrix and received vector can also be computed by concatenating the received signals and channel matrices:

$$\mathbf{H}_{CPD} = \begin{bmatrix} \mathbf{H}_1 \ \mathbf{H}_2 \\ \mathbf{H}_{rd} \ 0 \end{bmatrix}, \tag{5.25}$$

$$\mathbf{y}_{CPD} = \begin{bmatrix} \mathbf{y}_d^{(1)} \\ \mathbf{y}_d^{(2)} \end{bmatrix}. \tag{5.26}$$

After combining the effective $\mathbf{y}_{CPD}$ and $\mathbf{H}_{CPD}$, they are passed to a sphere detector to compute the LLR values and then passed to the channel decoder. Figure 5.4 summarizes these steps for cooperative partial detection.

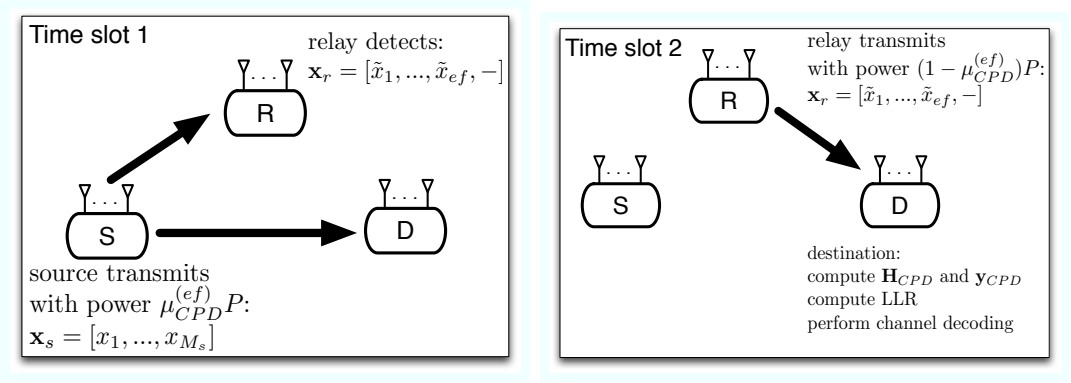

Figure 5.4 Cooperative Partial Detection (CPD) through MIMO relay node. In the first time slot, the relay receives a copy of the source multi-stream data, partially detects it, and forwards the detected data. In the second time slot, the receiver combines the multiple copies, as described earlier, to compute the LLR values. We denote the power splitting ratio by $\mu_{CPD}^{(ef)}$.

5.3 Computational Complexity Comparison

In this section, we derive and compare the complexity of the proposed techniques. The channel usually changes at a smaller rate than the received signal, and can be implemented with higher resource reuse in the hardware. Therefore, in computing the complexity, we mainly focus on the operations that happen in the symbol updating rate, as opposed to channel updating rate.

The complexity of a sphere detection operation, first presented in Chapter 2 for the point to point case and now applied to the relay scenario, can be derived from Eq. (5.11),

$$C_{SD}(M_s, w) = \sum_{i=M_s}^{1} C_i E\{D_i\}, \tag{5.27}$$

where C_i is the number of operations per node in the i-th level. In order to compute C_i, we refer to the VLSI implementation of [8], and note that, for each node, one needs to compute the $R_{i,j}b_j$ multiplications, where, except for the diagonal element, $R_{i,i}$, the rest of the multiplications are complex valued. The expansion procedure, Eq. (5.2), requires computing $R_{i,j}b_j$ for $j = i+1, ..., M_s$, which would require $(M_s - i)$ complex multiplications, and also computing $R_{i,i}b_i$ for all the possible choices of $b_j \in \mathcal{O}$. Even though there are w different b_j's, there are only $(\frac{\sqrt{w}}{2} - 1)$ different multiplications required for QAM modulations. For instance, for a 16-QAM system with $\{\pm 3 \pm 3j, \pm 1 \pm 1j, \pm 3 \pm 1j, \pm 1 \pm 3j\}$, computing only $(R_{i,j} \times 3)$ would be sufficient for all the choices of modulation points. Finally, computing the $\| \, . \, \|_2^2$ norm requires

a squarer or a multiplier, depending on the architecture and hardware availabilities.

In order to compute the number of adders for each norm expansion in (5.2), we assume a depth-first based tree search. Note that this assumption will not change the overall trends that we will observe later; however, it makes it possible to compute the complexity and compare the different scenarios. Therefore, there are $(M_s - i)$ complex valued adders required for $f_i{}' - \sum_{j=i+1}^{M_s} R_{i,j} b_j$, and w more complex adders to add the newly computed $R_{i,i} b_i$ values. Once the w different magnitudes, $|f_i{}' - \sum_{j=i}^{M_s} R_{i,j} b_j|^2$, are computed, they need to be added to the partial distance coming from the higher level, which requires w more addition procedures. Finally, unless the search is happening at the end of the tree, the norms need to be sorted, which assuming a simple sorter, requires $w(w+1)/2$ compare-select operations.

Therefore, keeping in mind that each complex multiplier corresponds to four real-valued multipliers and two real-valued adders, and that every complex adder corresponds to two real-valued adders, C_i is calculated by

$$\begin{aligned} C_i(M_s, w) &= \gamma((\frac{\sqrt{w}}{2} - 1) + 4(M_s - i) + 1) \\ &+ \theta(2(M_s - i) + 2w + w) \\ &+ \beta(w(w+1)/2) \cdot \text{step}(i-1), \end{aligned} \tag{5.28}$$

where $\text{step}(i-1)$ is used to ensure sorting is counted only when the search has not

reached the end of the tree, and is equal to:

$$\text{step}(t) = \begin{cases} 1, & t \geq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.29}$$

Moreover, we use θ, β and γ to represent the hardware-oriented costs for one adder, one compare-select and one multiplication operation, respectively. Based on FPGA and ASIC estimates, we choose $\theta = 1$, $\beta = 1$ and $\gamma = 10$ throughout this chapter.

We note that this is only one method of implementing this architecture, and depending on the architecture and timing requirements, other architectures could be used, which may lead to slightly different implementations and computation counts. However, these differences will not produce significant impact on our comparisons since our goal is to *compare* different cooperative schemes, assuming that all of them use the *same* MIMO detector structures.

Therefore, the computational complexity in the relay for the CPD is given by:

$$C_{relay}(M_s, ef, w) = C_{SD}(ef, w). \tag{5.30}$$

In order to compute the complexity in the destination, we extend the definition in (5.27) to soft sphere detectors that compute the LLR values for a list of size $|\mathcal{L}|$. The C_{SSD} is essentially similar to Eq. (5.27), except that $E\{D_i\}$ is now dependent

on the target list size:

$$C_{SSD}(M_s, w, |\mathcal{L}|) = C_{APP}(M_s, w, |\mathcal{L}|) + \sum_{i=M_s}^{1} C_i E\{D_i||\mathcal{L}|\} \quad (5.31)$$

where $C_{APP}(M_s, w, |\mathcal{L}|)$ is the number of operations required to compute the soft values (5.10).

Note that for similar M_s, w and list size $|\mathcal{L}|$, Eq. (5.10) remains the same. Therefore, for the sake of simplicity, we have not considered it in evaluating the C_{SSD}. Moreover, we use the concatenating methods of Eq. (5.8)-(5.9) and Eq. (5.25)-(5.26) in computing the destination complexity.

The total computational complexity in the destination for FDF and CPD are given, respectively, by:

$$\begin{aligned} C_{dest}^{(FDF)}(M_s, w) &= C_{SSD}(M_s, w, |\mathcal{L}|), \quad (5.32) \\ C_{dest}^{(CPD)}(M_s, ef, w) &= C_{SSD}(M_s, w, |\mathcal{L}|), \end{aligned}$$

where we ignore the slight computational difference in combining the received signals.

The simulation results for the complexity is shown in the next section.

5.4 Simulation Results

In this section, we compare the BER performance of the proposed detectors. We assume a three node relay network topology with the relay located between the source and destination, on the same line, and thus $d_{sd} = d_{sr} + d_{rd} = 1$. We further assume

that the path loss exponent is fixed to $\alpha = 3$. We fix the location of the relay, and then optimize the performance of the full detect-and-forward network by varying the power splitting ratio μ, as defined in Eq. (1.9), from the discrete set of $\{0.1, 0.2, ..., 0.9\}$, and call it μ_{FDF}. The power splitting ratio, μ_{FDF}, is the ratio of source transmit power to the total transmit power. In order to ensure that the savings in the relay are not limited to baseband processing saving, we also scale the transmit power of the relay by the ratio of the antennas being used. The power splitting ratio for the CPD case, $\mu_{CPD}^{(ef)}$, is, therefore defined as:

$$\mu_{CPD}^{(ef)} = 1 - (1 - \mu_{FDF})ef/M_s, \tag{5.33}$$

which implies that the relay transmit power in the CPD scenario is scaled down by a factor of (ef/M_s) compared to the FDF case, and the source uses a higher transmit power in return. This choice of transmit power allocation to relay and source better models the real-world per-antenna power constraint, and guarantees that by picking the partial detection strategy, the relay not only saves in the baseband computational processing, but also in the transmit power. For the sake of completeness, we also present the BER performance for a complete decode and forward scenario, where the relay fully detects and decodes the source signal, and then, re-encodes the signal and transmits that to the destination. The destination performs a process similar to the full detect-and-forward (FDF) case. Obviously, for the full decode-and-forward scenario, the complexity of the processing in the relay will be much higher than the

full detect-and-forward (FDF) case due to the full soft sphere detection and decoding process in the relay.

For this section's simulations, a rate 1/2 Turbo code is used in the source with an interleaver of size 1355, and feedback polynomial $(1 + D + D^2)$ and feedforward polynomial $(1+D^2)$. Rayleigh fading channel coefficients, as described in the previous sections of the chapter, are used.

Figure 5.5 shows the BER performance for a MIMO relay system with 4 antennas and a 16-QAM modulation. The relay is located at $d_{sr} = 0.2$ and the results are presented for different *ef* values. As *ef* increases, the performance gets closer to the Full Detect-and-Forward scenario. Therefore, the relay can adjust its level of complexity based on the available computational resources. The list size $|\mathcal{L}|$ is set to 100 in the destination for both the full detect-and-forward and full decode-and-forward. Note that since the relay is located relatively close to the source, it enjoys a very high SNR source-relay link, and therefore, performing the decoding procedure in the relay does not improve the performance significantly compared to just detecting.

Monte Carlo simulations are used to generate $E\{D_i\}$ and $E\{D_i||\mathcal{L}|\}$, and in combination with Eq. (5.28)-(5.33) compute the overall complexity for different total transmit power P values. Figures 5.6 and 5.7 show these results for a 4-antenna system with 16-QAM modulation in both the relay and the destination, assuming a list size $|\mathcal{L}| = 100$ in the destination. The relay requires less computational overhead

if it chooses to perform partial sphere detection with $1, 2$ or 3 streams of data. Both the FDF and CPD methods require a full sphere detection besides combinations of Eq. (5.8)-(5.9) and Eq. (5.25)-(5.26) at the destination. For very high SNRs, the average number of visited nodes per level equals one for the direct link scenario. Therefore, while the CPD and FDF methods need to perform different forms of cancellation and combining operations, the direct link scenario only performs sphere detection with, on average, one node per level.

Figures 5.8 and 5.9 show similar BER results for 3×3 and 4×4 systems with the relay located at $d_{sr} = 0.4$. The list size $|\mathcal{L}|$ is set to 60 in the destination for both the full detect-and-forward (FDF) and full decode-and-forward for the 3×3 case, and $|\mathcal{L}| = 100$ for the 4×4 case. Note that the gap between the full decode-and-forward and FDF is wider in $d_{sr} = 0.4$ cases compared to $d_{sr} = 0.2$ of Figure 5.5. This effect is due to the stronger channel between source and relay in the first case, i.e., $d_{sr} = 0.2$. In other words, since the source-relay channel is relatively stronger, the channel decoding in the relay does not improve the overall error performance, which is now dominated by other factors, such as the source-destination and relay-destination links.

In order to better understand the complexity-performance tradeoff, we present the minimum total transmit power P required to achieve a target BER. This is shown in Figure 5.10, where the required power is plotted versus the expansion factor, ef.

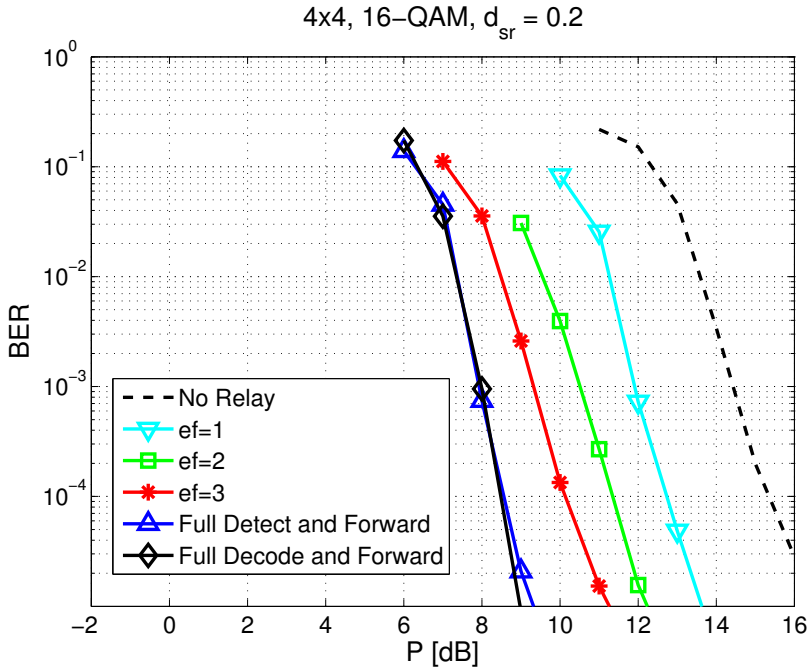

Figure 5.5 BER comparison for a system with $M_s = M_d = 4$ and 16-QAM. The relay is located at $d_{sr} = 0.2$. The power splitting ratios of the full detect-and-forward and full decode-and-forward is set to $\mu_{FDF} = 0.6$. The $\mu_{CPD}^{(ef)}$ for $ef = 3, 2$ and 1 is set to $0.7, 0.8$ and 0.9, respectively.

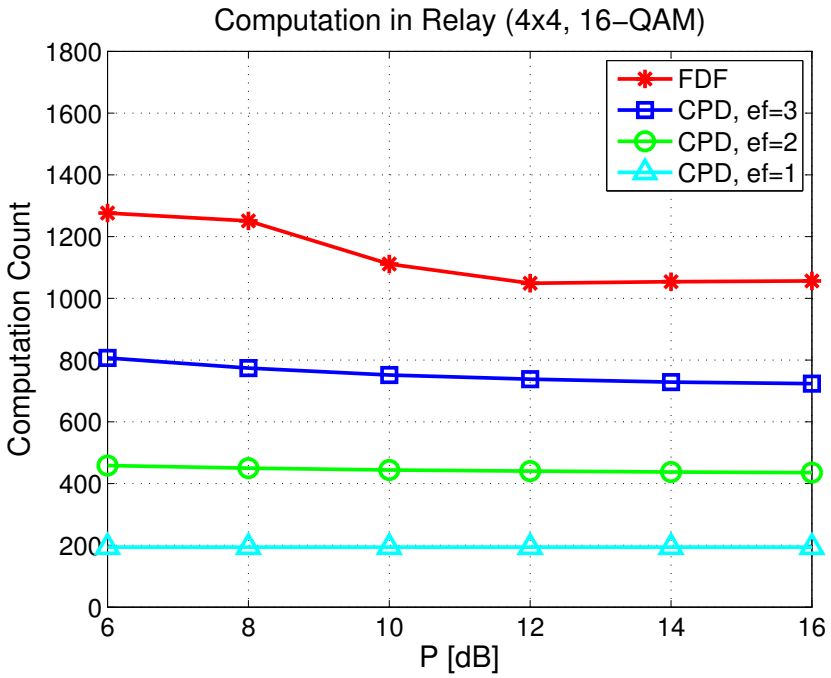

Figure 5.6 Comparison between the complexity of detection in relay for Full-Detect-and-Forward (FDF) and Cooperative Partial Detection (CPD) with expansion factors of 2 and 3. The relay is located at $d_{sr} = 0.2$. The power splitting ratios of the full detect-and-forward and full decode-and-forward is set to $\mu_{FDF} = 0.6$. The $\mu_{CPD}^{(ef)}$ for $ef = 3, 2$ and 1 is set to $0.7, 0.8$ and 0.9, respectively.

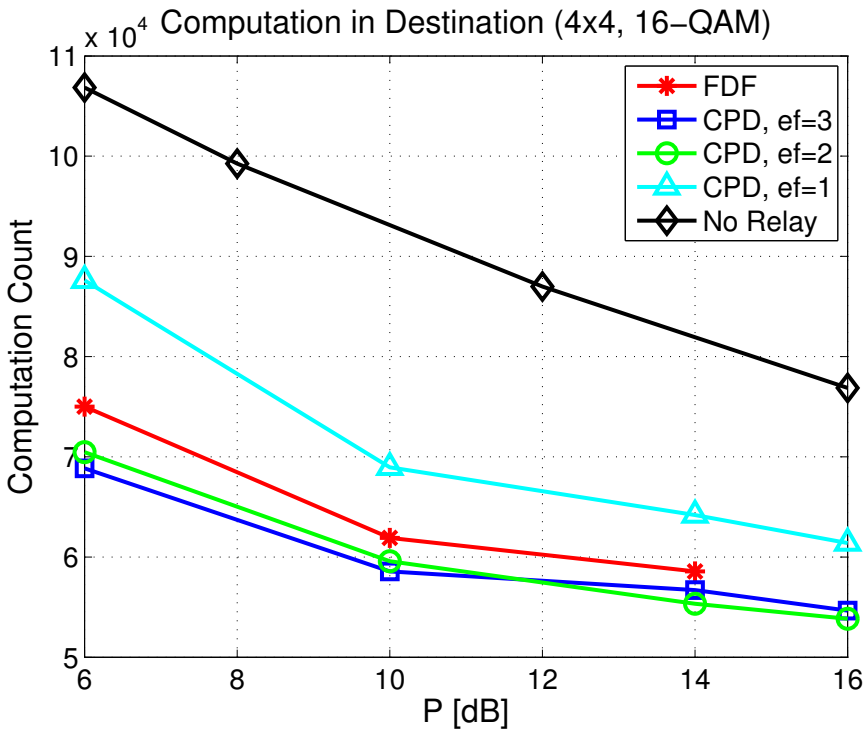

Figure 5.7 Comparison between the complexity of detection in the destination for Full-Detect-and-Forward (FDF) and Cooperative Partial Detection (CPD), with expansion factors of $1, 2$ and 3.

Detecting more streams in the relay, i.e., higher ef, improved the overall performance; therefore, higher ef translates into lower required power. Similar to the earlier simulation results, for each relay location, the power splitting ratio, μ_{FDF}, that achieves a better performance for the full detect-and-forward is picked from the limited set of $\{0.1, 0.2, ..., 0.9\}$. Then, the corresponding $\mu_{CPD}^{(ef)}$ for the partial detection schemes are chosen according to Eq. (5.33). Furthermore, the complexity is represented with the expansion factor ef.

5.4.1 Complexity Sensitivity

In the previous section, in computing the overall complexity, we assumed that the multiplier costs ten times more than the adder-subtractor modules. This assumption is based on typical ASIC results. However, changing these relative costs has minimal impact on the overall complexity behavior when we compare the complexity for different expansion factors, ef. In particular, as shown in Figures 5.11 and 5.12, changing the ratio of the multiplier to adder cost does not change the complexity trend for different ef values.

In order to better understand the relative changes in the computational complexity, Figure 5.13 shows a similar comparison, when the complexity of the $ef = 1$ case, i.e., the base case, is normalized to one. As the ratio of mul/add cost increases, the relative cost improvement due to using lower ef values increases. However, the complexity of $ef = 4$ case remains between 4.2 and 5.2 times the base case.

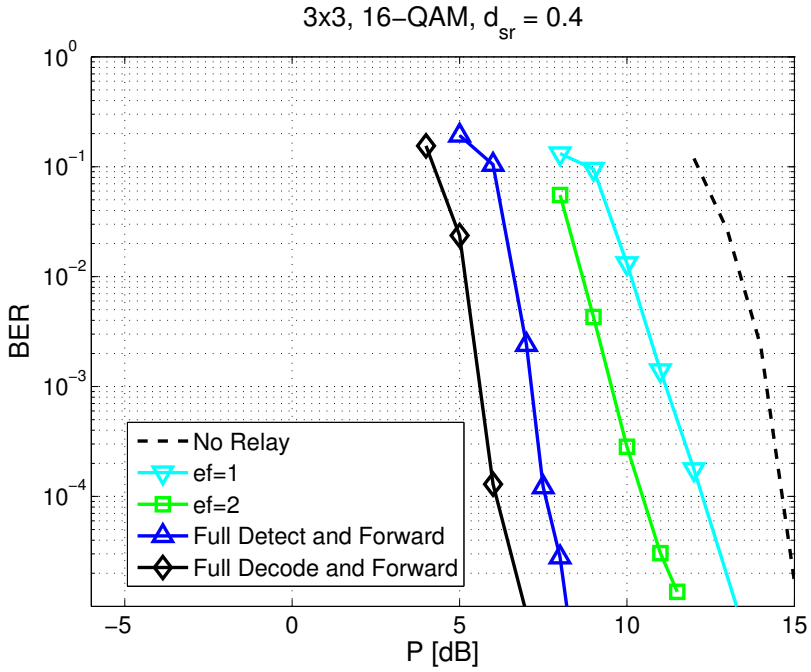

Figure 5.8 BER comparison for a system with $M_s = M_d = 3$ and 16-QAM. The relay is located at $d_{sr} = 0.4$. The power splitting ratios of the full detect-and-forward and full decode-and-forward is set to $\mu_{FDF} = 0.6$. The $\mu_{CPD}^{(ef)}$ for $ef = 2$ and 1 is set to 0.73 and 0.86, respectively.

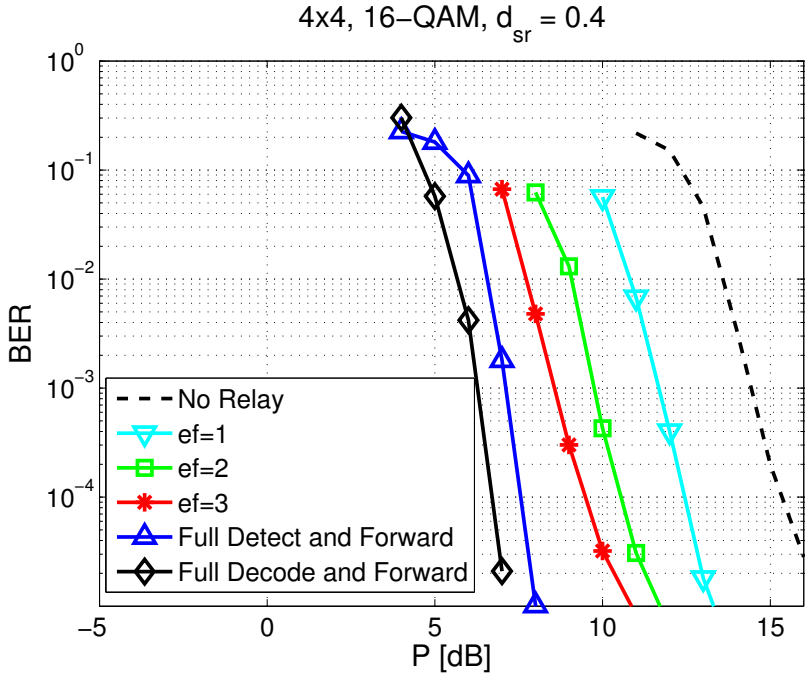

Figure 5.9 BER comparison for a system with $M_s = M_d = 4$ and 16-QAM. The relay is located at $d_{sr} = 0.4$. The power splitting ratios of the full detect-and-forward and full decode-and-forward is set to $\mu_{FDF} = 0.65$. The $\mu_{CPD}^{(ef)}$ for $ef = 3, 2$ and 1 is set to $0.7375, 0.825$ and 0.9125, respectively.

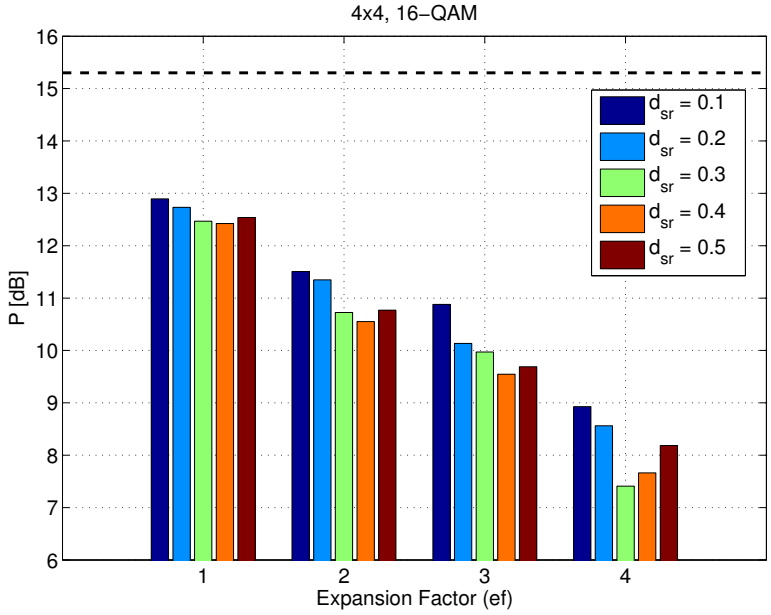

Figure 5.10 Performance-complexity tradeoff for a 4×4, 16-QAM system with relay located at $d_{sr} = 0.1, 0.2, 0.3, 0.4$, and 0.5. The vertical axis corresponds to the required total transmit power to achieve a BER of 10^{-4}, and the horizontal axis represents the expansion factor ef. The last set of bars, i.e., $ef = 4$, corresponds to the full-detect-and-forward, and the dashed line corresponds to the no relay scenario.

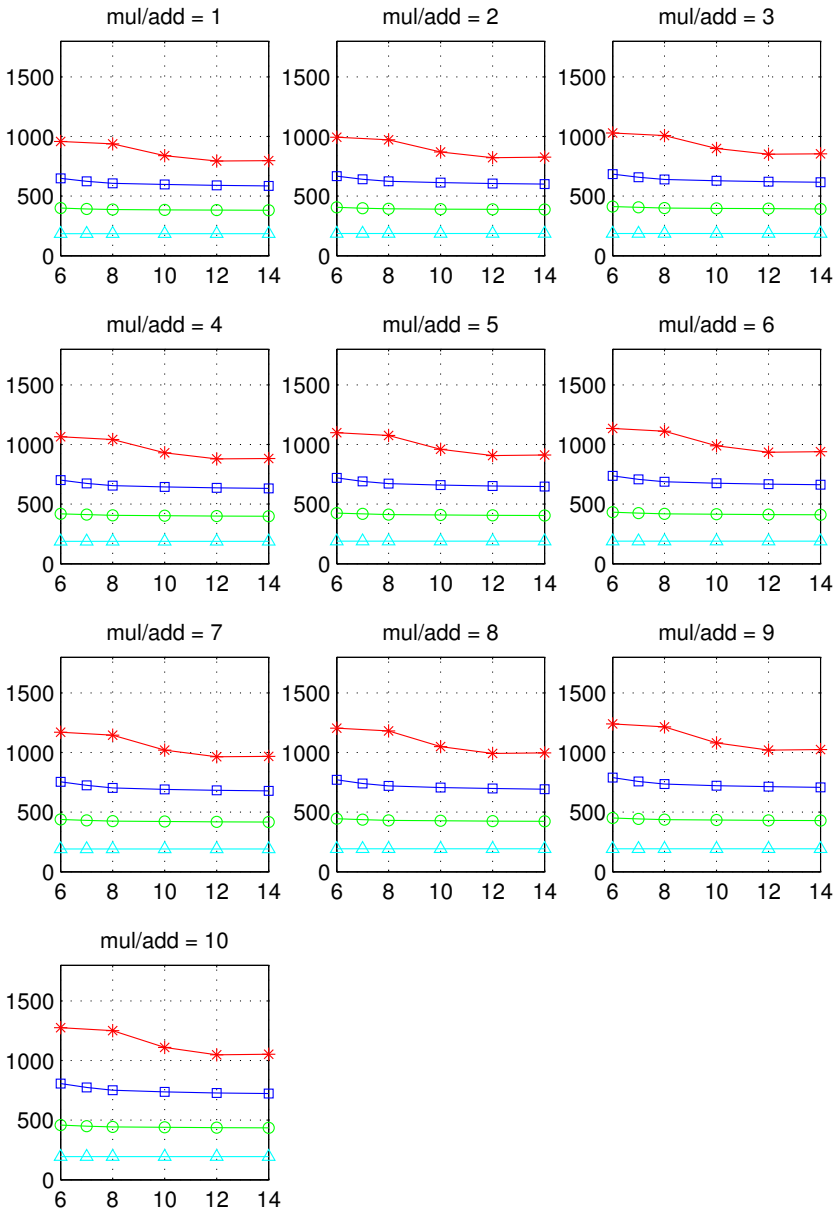

Figure 5.11 Complexity comparison for different ratios of the multiplier to adder cost. The vertical axes correspond to the computational complexity and the horizontal axes correspond to the total transmit power. Changing the ratio does not change the complexity trend for different *ef* values.

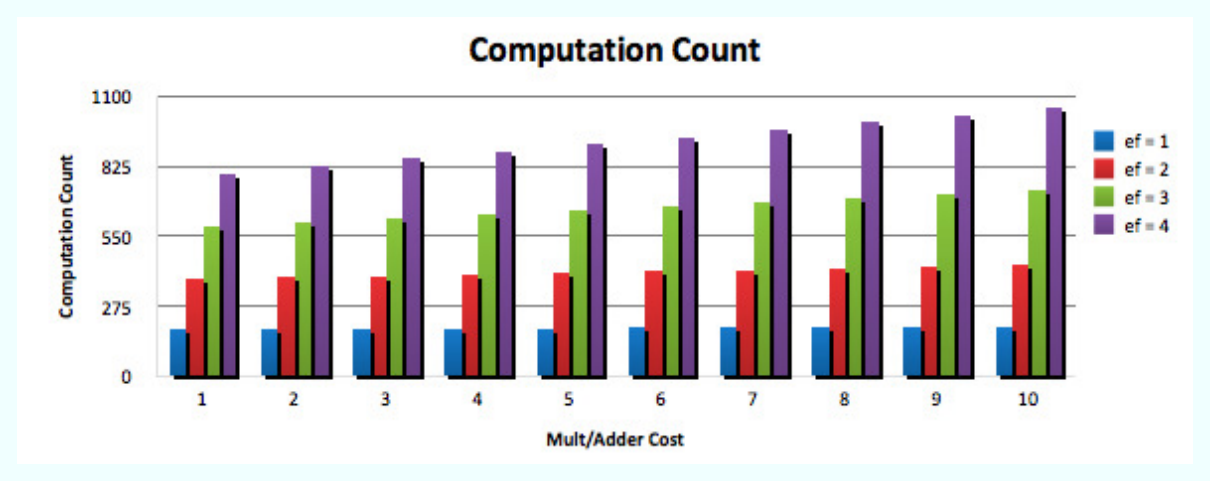

Figure 5.12 The computational complexity for different relative multiplier/adder costs for expansion factors of $ef = 1, 2, 3,$ and 4. The numbers correspond to the complexity when a total transmit power of $P = 12$ dB is used.

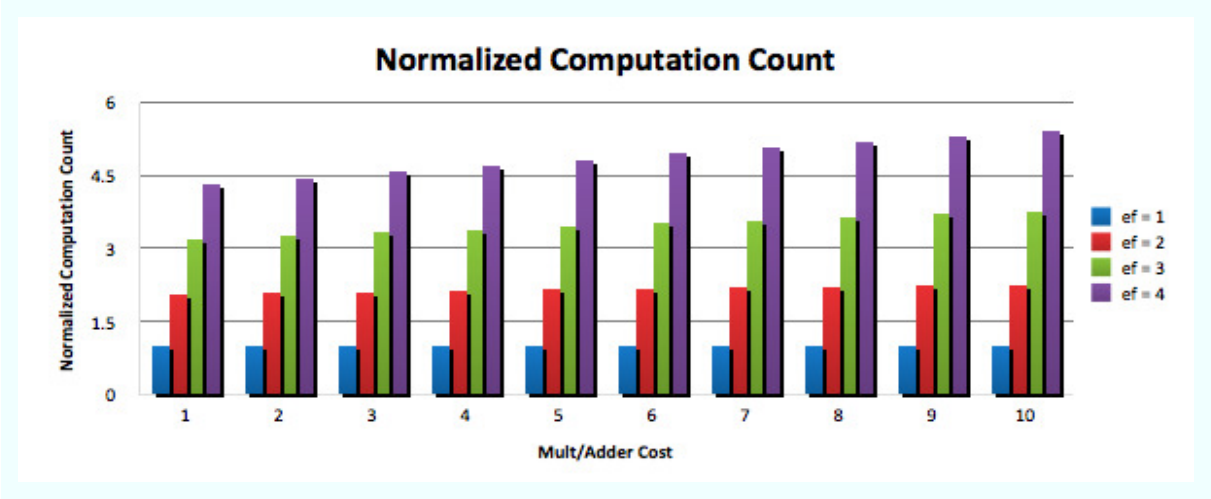

Figure 5.13 The computational complexity for different relative multiplier/adder costs, for expansion factors of $ef = 1, 2, 3$, and 4, when the computational complexity of the base case, i.e., $ef = 1$, is normalized to one. The numbers correspond to the complexity when a total transmit power of $P = 12$ dB is used.

5.4.2 Equal Rate Comparison

In this section, we present the BER performance comparison for the cases where the CPD and Direct Link scenarios use equal total transmission rate. For such cases, we assume similar modulation order for all the cases, but a coding rate of 1/4 for the direct link and a coding rate of 1/2 for the cooperation case. Therefore, the total spectral efficiency will be equal in all cases. Moreover, we assume that the transmit power from source remains the same in all cases, and the relay total transmit power is equal to $\frac{ef}{M_T}P_s$, where P_s is the total source transmit power. Note that with this power splitting assumption, the total transmit power over the two time slots is still less than or equal to the total transmit power in the no-relay case; therefore, this power increase still does not adversely affect the overall users' transmit power in the cell. Figures 5.14 and 5.15 shows this performance comparison when the relay is located at $d_{sr} = 0.4$ and $d_{sr} = 0.5$.

Note that the BER performance for both amplify-and-forward (AF) as well as the zero-forcing (ZF) in the relay are also presented for the sake of comparison. In the case of AF, the relay does not detect, and only amplifies the received signal and forwards that to the destination. Therefore, the destination needs to know all the channels, including the source-relay, $\mathbf{H}_{sr}$. Therefore, while the AF slightly improves the performance, it comes at a large price of transmitting source-relay channel state information alongside the data from relay to destination. In the ZF scenario, the

relay performs full detection using zero-forcing and forwards all the symbols to the destination using full transmit power of P_s. Note that with the exception of $ef = 1$, ZF performs worse than CPD even though it uses larger transmit power.

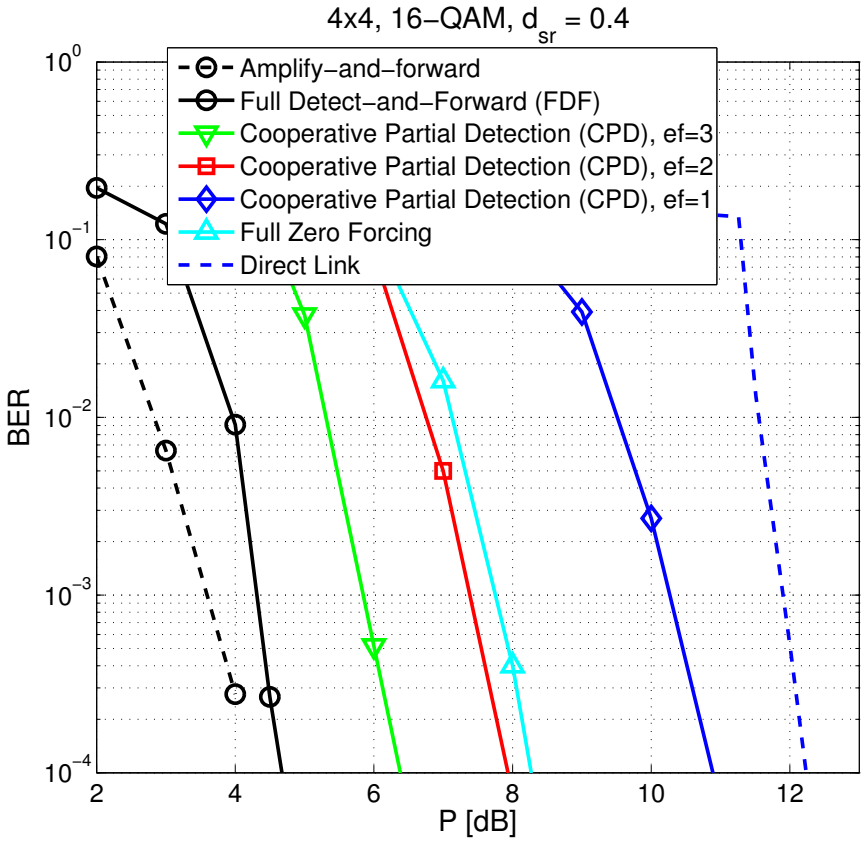

Figure 5.14 BER performance comparison with equal transmission rate for $d_{sr} = 0.4$.

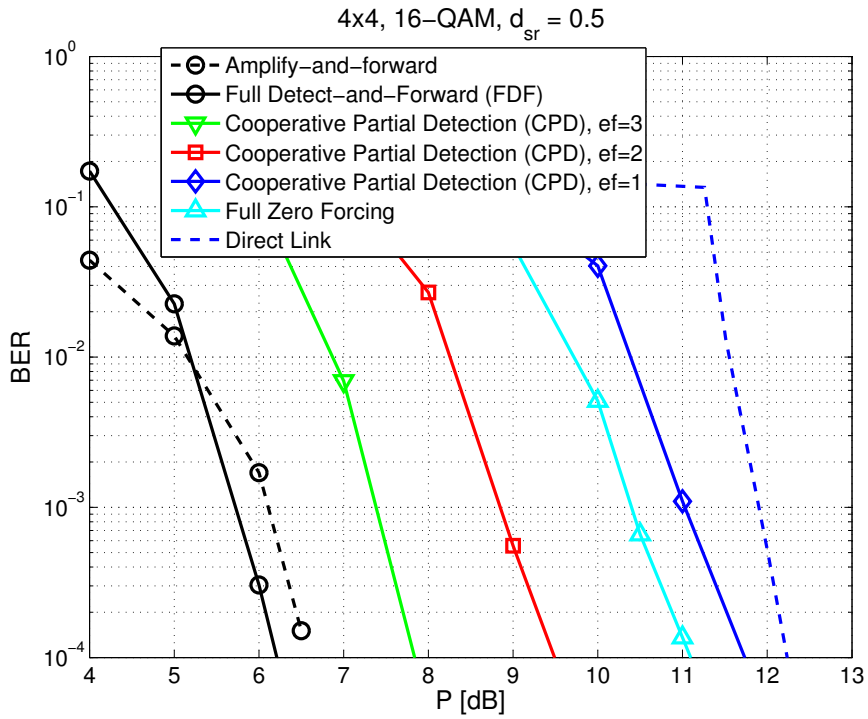

Figure 5.15 BER performance comparison with equal transmission rate for $d_{sr} = 0.5$.

5.5 Cooperative Partial Detection with K-best Detection

So far, we have used depth-first sphere detection to perform MIMO detection in the relay. The other very common MIMO detection, which is based on the breadth-first detection, is K-best MIMO detection. We first consider an uncoded K-best detection; then, we extend it to a system with channel coding.

5.5.1 Uncoded Detection

We will assume two transmission time slots: during the **first** time slot, the source transmits its vector message, $\mathbf{x}_s$ in Equations (1.1) and (1.2), to the relay and destination. The relay receives $\mathbf{y}_r$ of Eq. (1.1), and *partially* detects the message through the process described in section 5.2.1, i.e., it detects an ef-length subset of the elements of the transmit vector. In the **second** time slot, the relay forwards its detected vector of the first time slot, $\mathbf{x}_r$, to the destination. The detection procedure in the destination is comprised of four steps, as shown in Figure 5.16:

1. The destination, using the copy that it received from the relay in the second time slot, $\mathbf{y}_d^{(2)}$, detects the $\mathbf{x}_r$ vector, and calls this detected vector $\mathbf{x}_d^{(2)}$.

2. The $\mathbf{x}_d^{(2)}$ vector is cancelled from $\mathbf{y}_d^{(1)}$:

$$\mathbf{y}_d^{(tmp)} = \mathbf{y}_d^{(1)} - \mathbf{H}_{sd}^{(1:ef)}\mathbf{x}_d^{(2)}. \tag{5.34}$$

3. Using $\mathbf{y}_d^{(tmp)}$, the remaining streams, i.e., those not detected in the relay and thus not forwarded to the destination, are detected and called $\mathbf{x}_d^{(1)}$.

4. The two vectors, $\mathbf{x}_d^{(1)}$ and $\mathbf{x}_d^{(2)}$ are concatenated to form the final detected vector,

$$\mathbf{x}_d = [\mathbf{x}_d^{(2)}, \mathbf{x}_d^{(1)}]. \tag{5.35}$$

Note that the detection in the first and third steps, in the destination, are performed using the typical K-best detector; whereas, the relay performs the partial K-best method of section 5.2.1. Therefore, there are three major detection parameters which may be lumped together in the $\mathbf{k}_{CPD} = (K_r, K_{dr}, K_{ds})$ triple, where K_r is the K value for the partial K-best detection in the relay, and (K_{dr}, K_{ds}) are the K values for the K-best detection in the destination. K_{dr} is the K value for the K-best detection of $\mathbf{x}_r$ in step 1, and K_{sr} is the K value used for the detection of the remaining streams in step 3.

5.5.2 Coded Detection

In order to extend the partial cooperative detection to coded systems, we employ the QRD-M technique [108, 109] in the destination to compute the soft information required in the decoder. Therefore, while the relay performs the regular partial detection described in section 5.5.1, the destination performs a QRD-M detection in both of its internal detection steps. Therefore, the modified detection procedure, from section 5.5.1, in the destination is:

1. The destination, using the copy that it received from the relay in the second

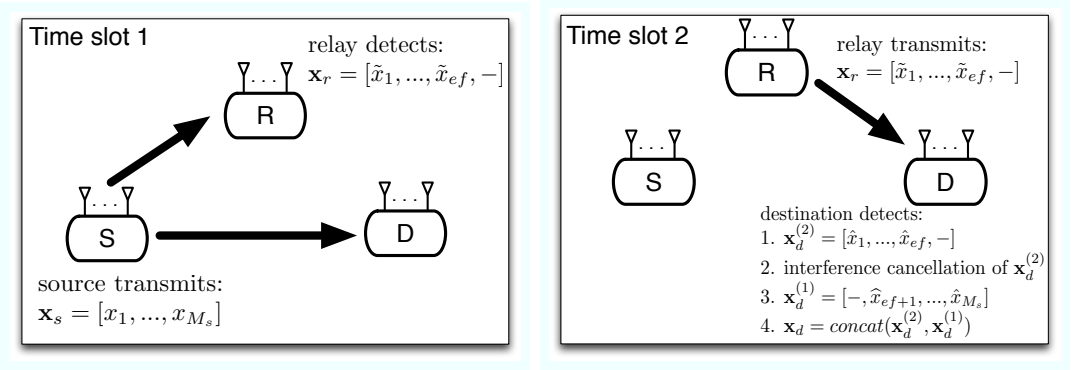

Figure 5.16 Cooperative partial detection through MIMO relay node in the K-best detection case. In the first time slot, the relay receives a copy of the source multi-stream data, partially detects it, and forwards the detected data. In the second time slot, the receiver first detects the copy received from the relay, then performs interference cancellation of the detected vector from the copy of the first time slot, and detects the remaining streams.

time slot, $\mathbf{y}_d^{(2)}$, detects the $\mathbf{x}_r$ vector, and calls this detected vector, $\mathbf{x}_d^{(2)}$. Moreover, using the list of the surviving candidates of the last level of the tree, $\mathcal{L}^{(2)}$, compute the corresponding LLR values $\mathbf{L}_E^{(2)}$.

2. The $\mathbf{x}_d^{(2)}$ vector is cancelled from $\mathbf{y}_d^{(1)}$:

$$\mathbf{y}_d^{(tmp)} = \mathbf{y}_d^{(1)} - \mathbf{H}_{sd}\mathbf{x}_d^{(2)} \tag{5.36}$$

3. Using $\mathbf{y}_d^{(tmp)}$, the remaining streams, i.e., those not detected in the relay and thus not forwarded to the destination, are detected and called $\mathbf{x}_d^{(1)}$. Then, using the list of the surviving candidates of the last level of the tree, $\mathcal{L}^{(1)}$, compute the corresponding LLR values $\mathbf{L}_E^{(1)}$.

4. Based on the power reordering of the streams that have been performed in the relay, the two vectors, $\mathbf{x}_d^{(1)}$ and $\mathbf{x}_d^{(2)}$, are concatenated to form the final detected vector,

$$\mathbf{x}_d = [\mathbf{x}_d^{(2)}, \mathbf{x}_d^{(1)}] \tag{5.37}$$

$$\mathbf{L}_E = [\mathbf{L}_E^{(2)}, \mathbf{L}_E^{(1)}]. \tag{5.38}$$

5.5.3 Complexity Analysis

In order to compare the complexity, we count the number of computations in the relay and destination. As discussed in [63, 64], the number of additions and compare-select operations to perform K-best is given by:

$$C_{add} = 2w' + 2w + 2Kw'(2m-2) + K(m(2m-1)-1), \tag{5.39}$$

$$C_{compare-select} = f_K(Kw')(2m-3) + f_K(w) + f_1(Kw'), \tag{5.40}$$

where $w' = \sqrt{w}$, and m is the number of transmit antennas, i.e., the transmit vector length. Also, we define $f_K(l) = K \times (l - K)$ where $f_K(l)$ is the number of compare-select operations required for finding the best K candidates in a list of size l if *bubble sort* is used. It was shown in [79] that bubble sort is a suitable VLSI implementation choice for a wide range of K-best detectors as long as K is larger than w'. Note that (5.39) and (5.40) do not account for the ordering and QR decomposition, as those operations take place at a much slower rate compared to the core K-best detector unit. Also, the multiplications are replaced with shift-add operations as one term of the products are scaled integers of the modulation set.

Knowing the number of computations for addition and compare-select operations from Eq. (5.39) and (5.40), we can write the overall number of computations for one K-best detection as a function of m, K, w:

$$C_{K-best}(m, w, K) = \theta \cdot C_{add} + \beta \cdot C_{compare-select}, \tag{5.41}$$

where we use θ and β to represent the hardware-oriented costs for one real-valued adder and one real-valued compare-select operation, respectively.

Having derived the overall computation cost of the generic K-best detector of

Eq. (5.1), we can compute the computation cost of the cooperative detection scheme of section 5.5. Therefore, given the parameters of the cooperative system, the numbers of operations required to perform the proposed cooperative partial detection in the relay and destination, are

$$\begin{aligned} C^{R}_{CPD}(M_s, w, \mathbf{k}_{CPD}) &= C_{K-best}(ef, K_r, w), \\ C^{D}_{CPD}(M_s, w, \mathbf{k}_{CPD}) &= C_{K-best}(ef, K_{dr}, w) \\ &+ C_{K-best}(M_s - ef, K_{ds}, w) \\ &+ C_{LLR}(M_s, K'_d, w) \\ &+ C_{LLR}(M_s - ef, K'_d, w), \end{aligned} \tag{5.42}$$

where C^{R}_{CPD} is the computation cost of the detection in the relay, i.e., a partial K-best detection with K_r. The C^{D}_{CPD} represents the computation cost in the destination, which comprises two consecutive K-best detection stages, each followed by an LLR calculation block, based on Eq. (5.10).

The LLR value computation count is given by

$$C_{LLR}(M_s, K, w) = (\log_2 w) \cdot M_s \cdot \{\gamma + 2\beta f_1(Kw')\},$$

where we use γ and β to represent the hardware-oriented costs for one real-valued multiplier and one real-valued compare-select operation, respectively.

Furthermore, for full detect-and-forward (FDF), which is based on full symbol-level detection of all the antenna streams in the relay and a symbol-level maximal

ratio combining in the destination followed by a K-best detection, the computation costs in the relay and destination are given by

$$\begin{aligned} C^{R}_{FDF}(M_s, w, \mathbf{k}_{FDF}) &= C_{K-best}(M_s, K'_r, w), \\ C^{D}_{FDF}(M_s, w, \mathbf{k}_{FDF}) &= C_{K-best}(M_s, K'_d, w) \\ &+ C_{MRC}(M_s) \\ &+ C_{LLR}(M_s, K'_d, w), \end{aligned} \tag{5.43}$$

where the first terms on the right hand side (RHS) of both equations correspond to the detection in the relay and destination respectively. The K-values for the full K-best detection steps in the relay and the destination are denoted by $\mathbf{k}_{FDF} = (K'_r, K'_d)$. The C_{MRC} and C_{LLR} values represent the computational cost of computing the maximal-ratio combining (MRC) and LLR values. The C_{MRC}, which consists of two complex-valued matrix-by-vector multiplications and one complex-valued matrix addition, is given by

$$C_{MRC}(M_s) = \theta\{2(2M_s^2 + 2M_s(M_s - 1) + M_s)\} + \gamma\{2 \times 4M_s^2\},$$

where θ and γ represent the hardware-oriented costs for one real-valued adder and one real-valued multiplication operation, respectively.

The complexity of the relay, C^{R}_{CPD}, using the proposed CPD, for different values of expansion factor and modulation orders, is shown in Figure 5.17. A 4-antenna system of source, relay and destination is assumed. Note that the final point, for

$ef = 4$, corresponds to the full detect-and-forward computation count, C^R_{FDF}. From this figure, it is clear that performing a partial detection, as opposed to full detection, can save significant computational processing in the relay.

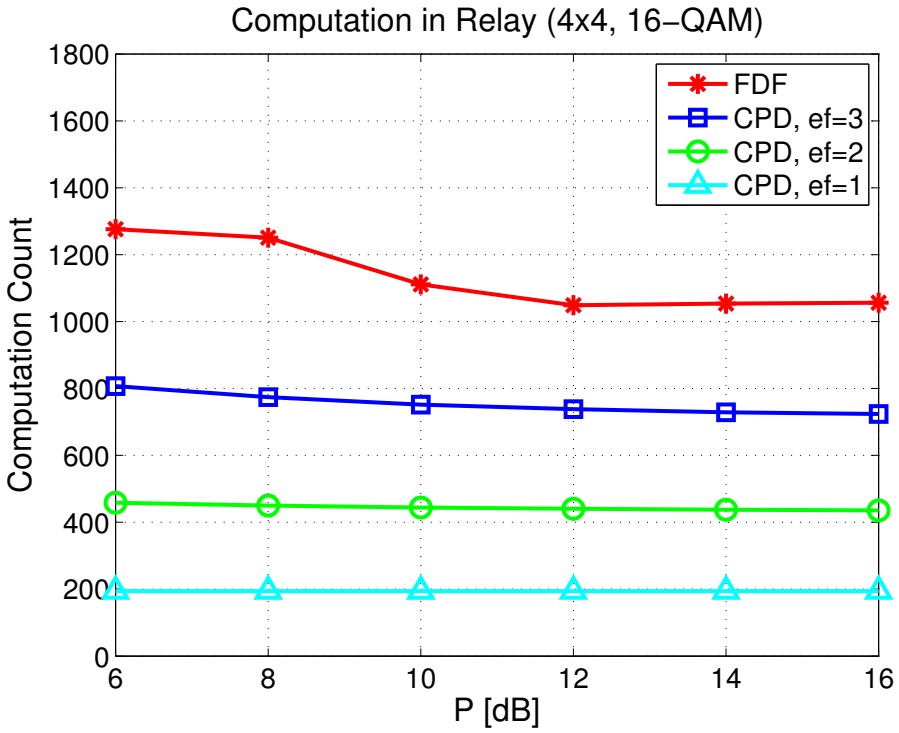

Figure 5.17 The relay complexity versus the expansion factor for $K_r = 5$ for Cooperative Partial Detection using K-best algorithm. The adder, compare-select and multiplier costs are assumed to be $\theta = \beta = 1$ and $\gamma = 10$.

5.5.4 Simulation Results

In this section, we present the simulation results for the proposed cooperative partial detection scheme. We show the BER performance improvement for different scenarios. Throughout this section, we set the detection parameters, namely K values for the K-best and partial K-best detections, in such a way that the direct-link has similar computational complexity as the computation of the destination in the cooperation mode. In other words, using the results of section 5.5.3, we set K_{dl}, $\mathbf{k}_{FDF} = (K'_r, K'_d)$ and the $\mathbf{k}_{CPD} = (K_r, K_{dr}, K_{ds})$ triple such that

$$\begin{aligned} C^D_{CPD}(M_s, w, \mathbf{k}_{CPD}) &= C^D_{FDF}(M_s, w, \mathbf{k}_{FDF}) \\ &= C_{K-best}(M_s, w, K_{dl}), \end{aligned} \tag{5.44}$$

where K_{dl} is the K value for the K-best detection in the direct-link scenario. Eq. (5.44) guarantees a fair comparison between the non-cooperation case, the full detect-and-forward case, and the proposed cooperative partial detection scheme.

In terms of the network topology, we assume a three node relay network with the relay located between the source and destination, on the same line, and thus $d_{sd} = d_{sr} + d_{rd}$. We further assume that the path loss exponent is fixed to $\alpha = 3$. The hardware-oriented complexity parameters for the real-valued adder, compare-select and multiplier are set to $\theta = \beta = 1$ and $\gamma = 10$. A rate $R = 1/2$ turbo code [2] with memory length of 2 and the generating polynomials of $G_1(D) = 1 + D + D^2$ and

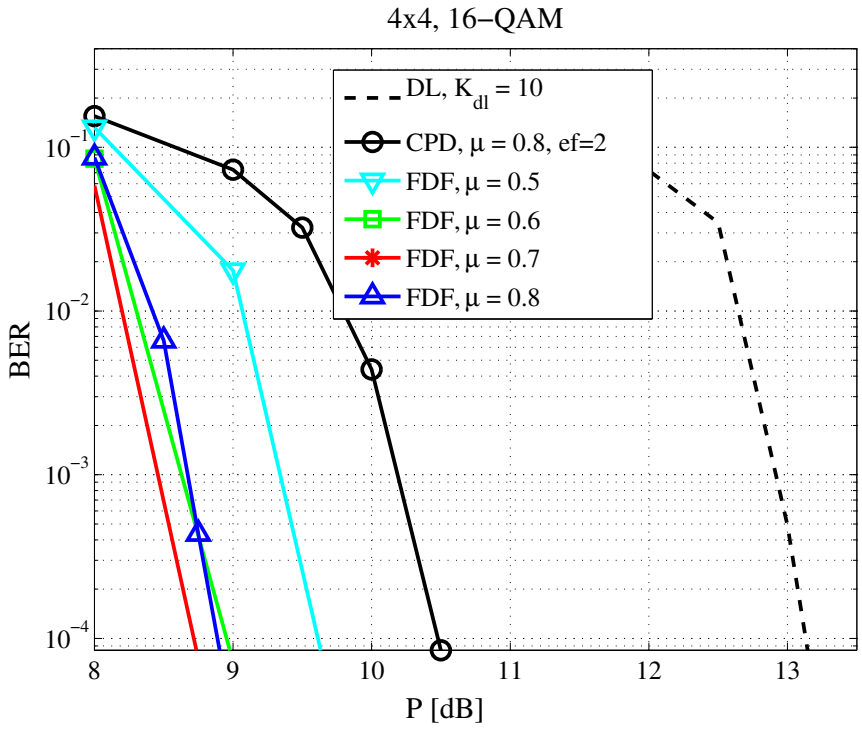

Figure 5.18 BER of 4×4, 16-QAM system with the relay located at $d_{sr} = 0.4$ from the source, and for different power splitting ratios, $\mu = 0.5, ..., 0.8$, between the source and the relay for the FDF case. Setting $\mathbf{k}_{CPD} = (5, 13, 13)$, $K_{dl} = 10$ and $\mathbf{k}_{FDF} = (5, 7)$ guarantees equal computational complexity for all the three scenarios in the destination. For the cooperative partial MIMO detection, the expansion factor is set to $ef = 2$; therefore, the relay detects and forwards 2 of the streams to the destination, and the destination detects those two streams from the relay followed by the remaining two using the original vector it received from the source.

$G_2(D) = 1 + D^2$ is used. The maximum number of inner iterations inside the turbo decoder is set to 8, and the detector to decoder is a single phase iteration, i.e., no feedback from the decoder to the soft detector. The information message length for each frame is of size 9355 bits.

Figure 5.18 shows the performance improvement due to using the proposed cooperative partial detection scheme, where the relay node is located in the first half of the line connecting the source and the destination, $d_{sr} = 0.4$. In order to guarantee the equal computation constraint of Eq. (5.44), we set $\mathbf{k}_{CPD} = (5, 13, 13)$ for the cooperative partial MIMO detection and $K_{dl} = 10$ for the direct link. Two of the streams are detected in the relay, i.e., $ef = 2$. Furthermore, in order to provide another comparison point, the BER of full detect-and-forward of a relay in the same location with varying transmit power ratio, μ, is presented as well. Note that for the full detect-and-forward case, the K values in the relay and the destination are set to $\mathbf{k}_{FDF} = (5, 7)$ in order to guarantee that the full detect-and-forward scheme undertakes similar computation overhead in the destination.

Note that the proposed cooperative partial detection scheme offers a 2.5 dB performance improvement over the non-relay, i.e. direct link, scenario, with limited complexity overhead. As shown in the previous section, the relay would have required three times higher complexity to achieve this performance. Figure 5.19 shows a similar behavior for a 5×5 system, and a relay detecting only $ef = 3$ streams out

of the 5 streams.

5.6 Destination to Relay Feedback

We assume that the destination orders the columns of the channel matrix $\mathbf{H}_{sd}$ using the sorted QR ordering of [30], and calls $\mathbf{P}$ the ordering array, where the first element of the array, $\mathbf{P}(1)$, has the weakest effective SNR, and the last element of the array, $\mathbf{P}(M_s)$, has the highest effective SNR. Then, it uses a feedback link to let the relay know of the index of the ef weakest antennas, $\mathbf{P}(1, ..., ef)$. The number of feedback bits is M_s, which is essentially a vector of 0s and 1s, where 1 means that the corresponding stream should be detected, and 0 means that the corresponding stream should not be detected. Note that since the channel changes at a slower rate compared to the transmitted data, this feedback link is a lower rate feedback link.

We will also assume two transmission time slots for the actual data: during the first time slot, the source transmits its vector message, $\mathbf{x}_s$, to the relay and destination. The relay receives $\mathbf{y}_r$, and partially detects the message through the process described in the previous sections, i.e., it detects an ef-length subset the elements of the transmit vector. However, the major difference is that the relay uses the antennas indexed in the $\mathbf{P}(1, ..., ef)$ array for the detection process. In other words, the detected antennas are the ones on the weakest effective SNR at the destination.

In the second time slot, the relay forwards its detected vector of the first time slot, $\mathbf{x}_r$, to the destination. The detection procedure in the destination is based on

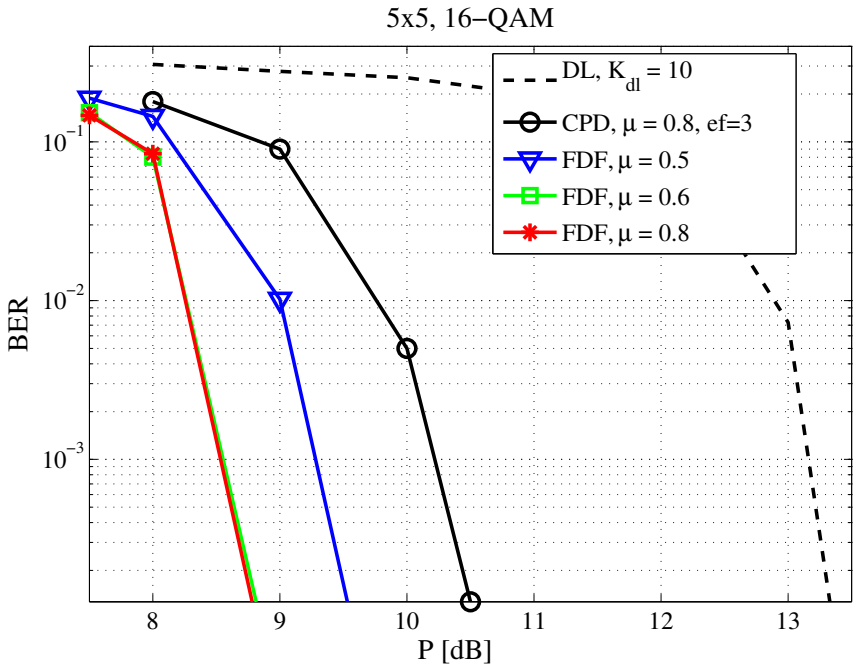

Figure 5.19 BER of 5×5, 16-QAM system with the relay located at $d_{sr} = 0.4$ from the source, and for different power splitting ratios, $\mu = 0.5, ..., 0.8$, between the source and the relay for the FDF case. Setting $\mathbf{k}_{CPD} = (5, 13, 13)$, $K_{dl} = 10$ and $\mathbf{k}_{FDF} = (5, 7)$ guarantees equal computational complexity for all three scenarios in the destination. For the cooperative partial MIMO detection, the expansion factor is set to $ef = 3$; therefore, the relay detects and forwards 3 of the streams to the destination, and the destination detects those three streams from the relay followed by the remaining two using the original vector it received from the source.

combining the received signals from the relay and source and performing a sphere detection on those. The simulation results for this feedback algorithm for an uncoded system is presented in Figures 5.20 and 5.21.

5.7 Managing Complexity in the Relay

We have, so far, demonstrated a variety of techniques to control the complexity in the relay. We are now going to formalize that, and provide insights that shows the steps the relay has to take. In essence, depending on the available resources in the relay as well as the target BER performance in the destination, the relay could choose how to cooperate with the source-destination link. In other words, the relay has a number of detection and transmission options. In the following paragraphs, we discuss the tradeoffs of every one of these strategies. Note that these comparisons are based on the results presented in the earlier sections of this chapter and chapter 4:

Full MIMO Detection (FDF): Performing full close-to-ML MIMO detection, e.g., different variations of sphere detection, in the relay and then transmitting all the received signals improves the performance significantly. However, that comes at a price of high relay complexity as shown earlier.

Cooperative Partial MIMO Detection (CPD): In this scenario, the relay only takes advantage of a subset of its antennas for transmission. In the trivial case of single-antenna relay, this leads to the relay using its only antenna for reception and transmission. However, in the more general case of multi-antenna relays, this would

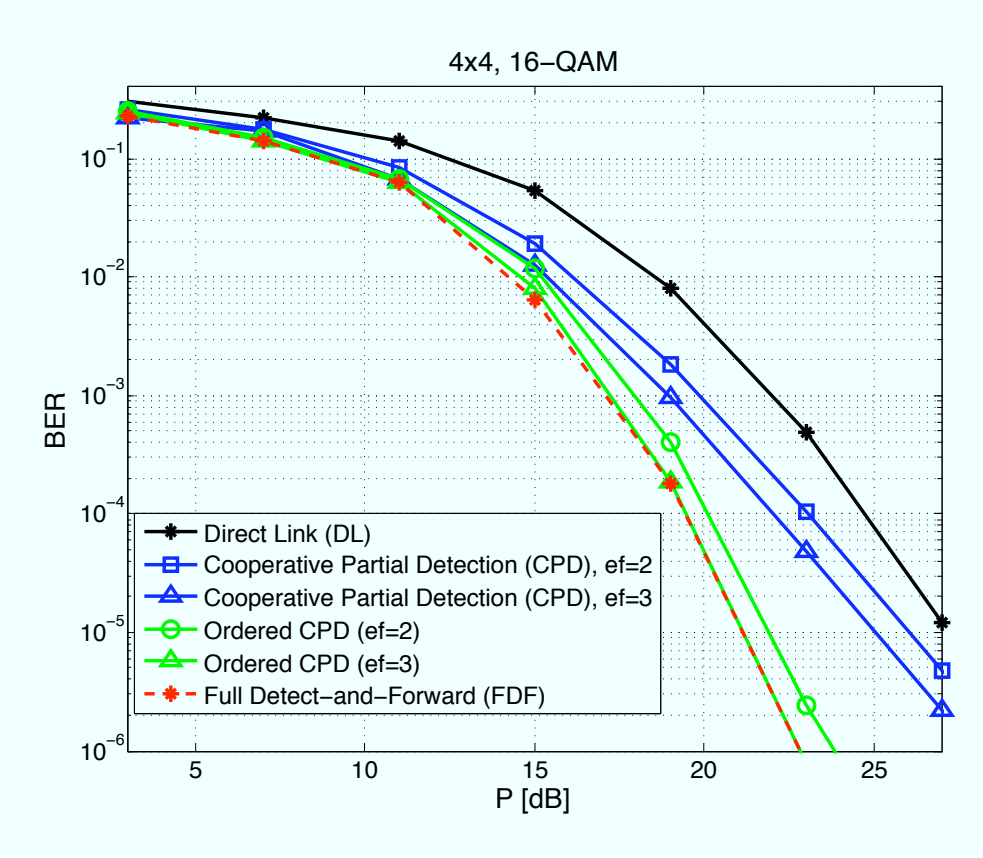

Figure 5.20 BER of 16-QAM system with 4 antennas, and the relay located at $d_{sr} = 0.2$. The expansion factor is set to $ef = 2$ and 3; therefore, the relay decodes either two or three of the signals, re-encodes that part, and sends that, along with the detected symbols of the remaining part. The power splitting ratios of the full detect-and-forward scheme is set to $\mu = 0.7$. The μ for $ef = 3$ and 2 is set to 0.775 and 0.85, respectively. The destination combines the transmissions it had received from the source and relay, and performs a full decoding. Rayleigh fading channels are assumed. The proposed ordered partial detection schemes, denoted by circled pattern curves, improve the performance significantly compared to the conventional CPD, shown with square and triangle patterns.

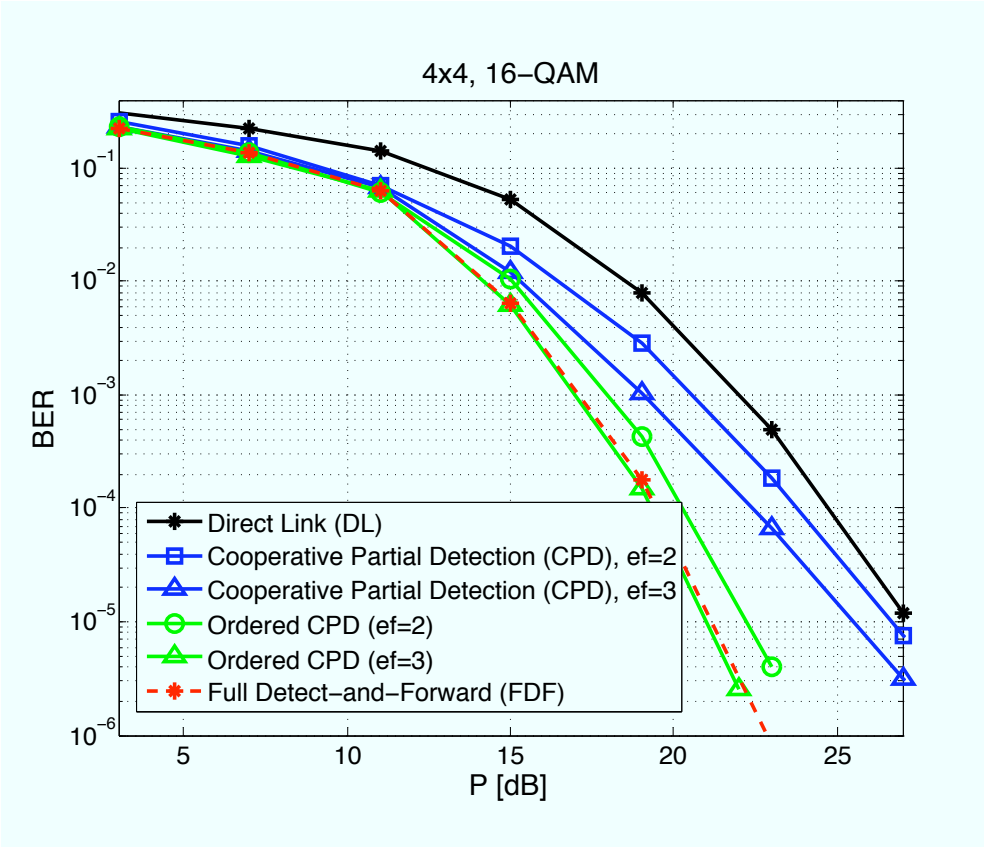

Figure 5.21 BER of 16-QAM system with 4 antennas, and the relay located at $d_{sr} = 0.2$. The expansion factor is set to $ef = 2$ and 3; therefore, the relay decodes either two or three of the signals, re-encodes that part, and sends that, along with the detected symbols of the remaining part. The difference between this set of results and those presented in Figure 5.20 is that in the current figure, the power splitting ratios of the full detect-and-forward scheme as well as all the CPD cases, i.e., $ef = 3$ and 2, is set to $\mu = 0.7$. The destination combines the transmissions it had received from source and relay, and performs a full decoding. Rayleigh fading channels are assumed. The proposed ordered partial detection schemes, denoted by circled pattern curves, improve the performance significantly compared to the conventional CPD, shown with square and triangle pattern curves.

lead to the relay using only a limited number of its antennas for transmission. The advantage of this technique is that the relay not only saves in the baseband processing, it also saves transmission power by choosing not to use all of its antennas. Note that we are making the practical assumption that the power per antenna is limited.

Linear Detection in the Relay: In this case, the relay chooses to perform a low-complexity linear detection, such as MMSE or Zero-forcing. While this technique can save some of the baseband processing in the relay, it still requires full transmission power as the relay has already detected all the antennas and transmitting only a subset of the detected streams leads to significant performance loss in the destination. Therefore, while this technique saves baseband processing power, it still has to use all of its transmission power in order to avoid high performance loss. Linear detection, therefore, can only be useful in comparison to very low expansion factor efs of CPD.

Amplify and Forward (AF): The AF technique can be fairly impractical in high modulation order and/or multi-antenna scenarios as it would require forwarding not only the amplified received signal, but also the channel matrix to the destination. While in the case of very low modulation orders or single-antenna scenarios, AF can be achieved without forwarding this information, in the more complex scenarios of multi-antenna systems, or even single-antenna and large modulation order systems (i.e. 16-QAM and higher), it fails to provide reasonable BER performance in the destination. Therefore, the performance improvement of the AF technique comes

at a rather large overhead of the source-relay channel matrix information in the destination. This overhead can lead to significant throughput loss.

We, now, provide comparisons in Figure 5.22 to demonstrate how these different techniques can perform under different resource-constraints in the relay.

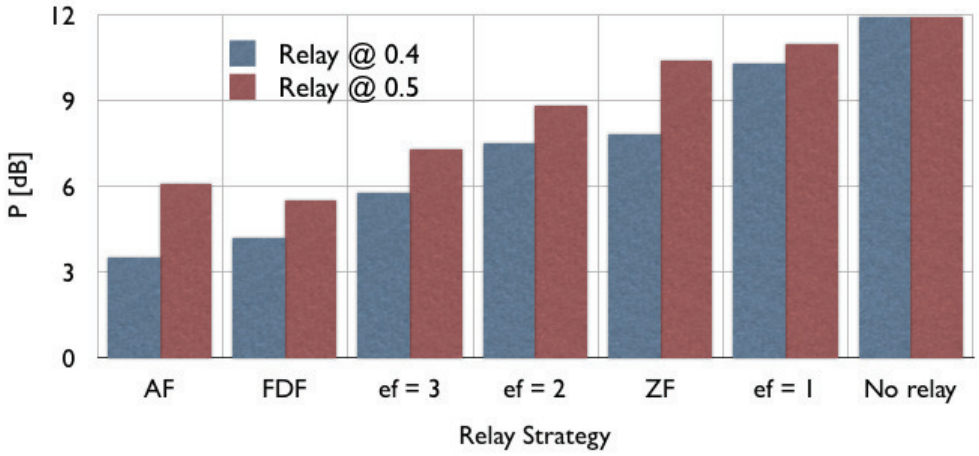

Figure 5.22 The complexity/performance tradeoff comparison that allows relay to choose what strategy to use for cooperation for relay distances, d_{sr}, of 0.4 and 0.5. Note that Amplify and Forward (AF) comes at a very high price of SR channel state information at the destination.

Chapter 6
Hardware Verification Using the WARP Platform

In this section, we describe the hardware platform to perform cooperative communication tests. We used the Wireless Open-access Research Platform (WARP) [4] to perform the hardware experiments of the cooperative partial detection algorithm. WARP hardware consists of a standalone main board with a Virtex-2 Pro or Virtex 4 FPGA, and four daughtercard slots. Each daughtercard slot can allow a custom radio board allowing for up to 4 antennas per board. Moreover, the FPGA boards can be stacked and connected through Multi-Giga Bit Transceivers (MGTs) allowing for even higher number of antennas per node. The daughtercard slots can be used for other applications, such as debugging boards, etc. A figure of the FPGA board along with the radio boards are shown in Figure 6.1.

WARP allows for real-time over-the-air transmission and reception of RF signals, and currently a custom Physical and MAC layer is available at [4].

6.1 Experiment Setup

WARPLab is a platform for rapid prototyping of the CPD physical layer algorithm over the air [62]. It takes advantage of the WARP hardware [19] and MATLAB at the same time. In WARPLab, WARP boards that correspond to the transmitter and receivers are connected to a PC through an Ethernet switch. Then, the raw I/Q

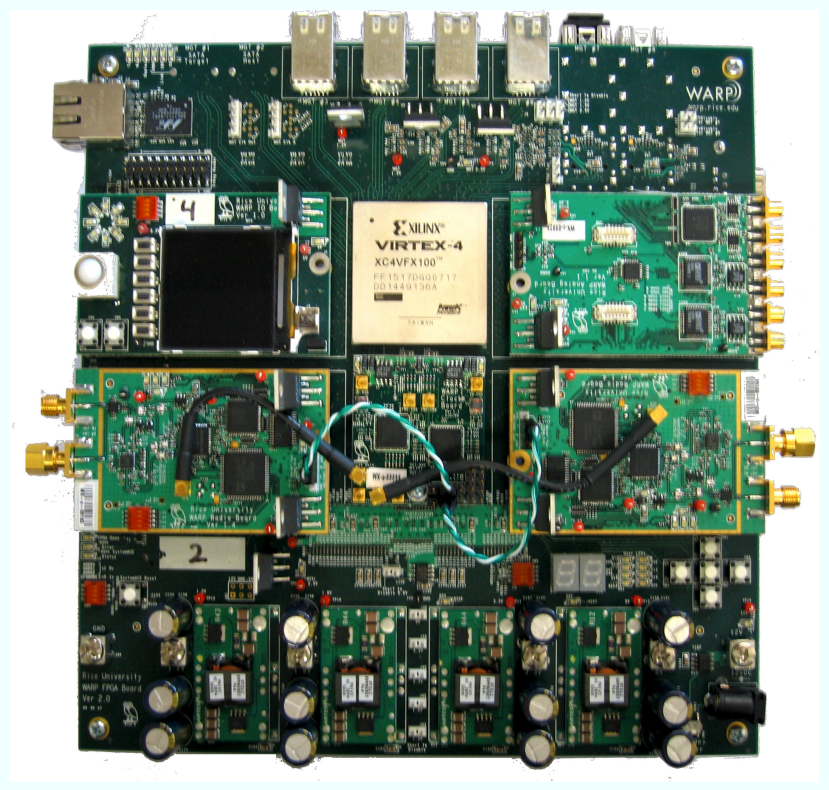

Figure 6.1 The next generation WARP board with four daughtercard slots. The board can support up to four radio daughtercards.

samples are generated in MATLAB and sent to the transmit WARP boards through Ethernet. The transmit boards will then modulate and transmit the raw samples to the 2.4 GHz frequency radio boards. The receive boards receive the RF signals, downconvert them to the baseband samples and send them back to MATLAB via Ethernet. Figure 6.2 shows the setup and connection to the channel emulator.

6.1.1 Azimuth Channel Emulator

Figure 6.2 shows our 2×2 three node cooperative system setup. The three WARP boards are connected to a PC through Ethernet. In order to emulate channel behavior, an Azimuth ACE 400 WB wireless channel emulator [1] is used. The emulator can support a range of channel models, including the TGn models as well as a subset of Winner channel models [5]. Moreover, custom channel models can be defined and emulated using the emulator's software interface.

The emulator can support up to two 4-antenna boards, which allows for a maximum of 2×2 MIMO relay experiment. Therefore, for the 2×2 full MIMO relay setup, we use 2 inputs, 4 outputs and 8 paths as shown in Figure 6.3. For the first time slot, we designate one node as the source, one node as the relay, and one node as the destination. In the second time slot, we designate one node as the relay and one node as the destination and connect the two nodes with four reverse paths.

The output transmit power of the WARP radio boards is set to 40, and the input power of the emulator ports are accordingly fixed using the WARPLab continuous

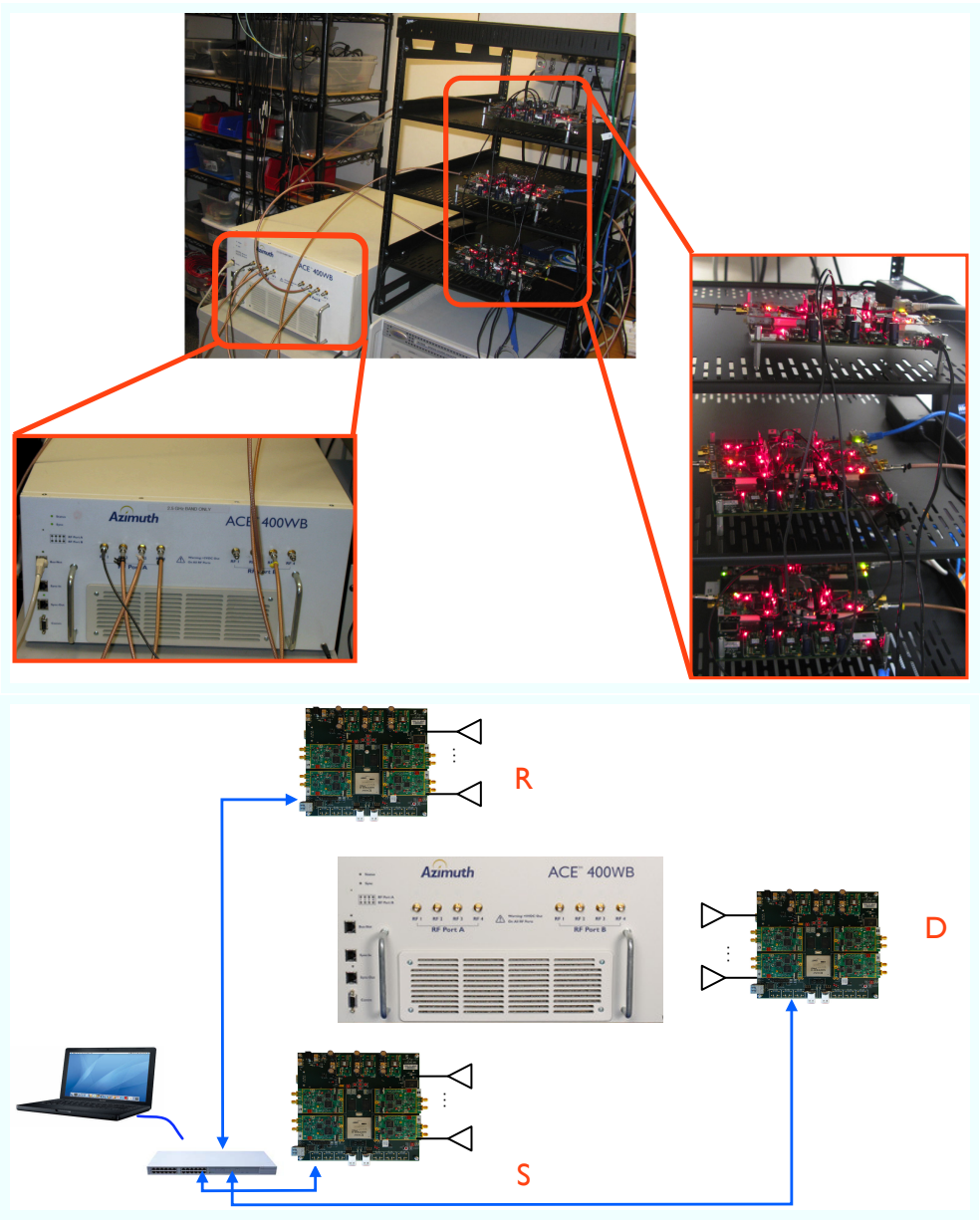

Figure 6.2 Test setup using the WARP boards and the Azimuth channel emulator.

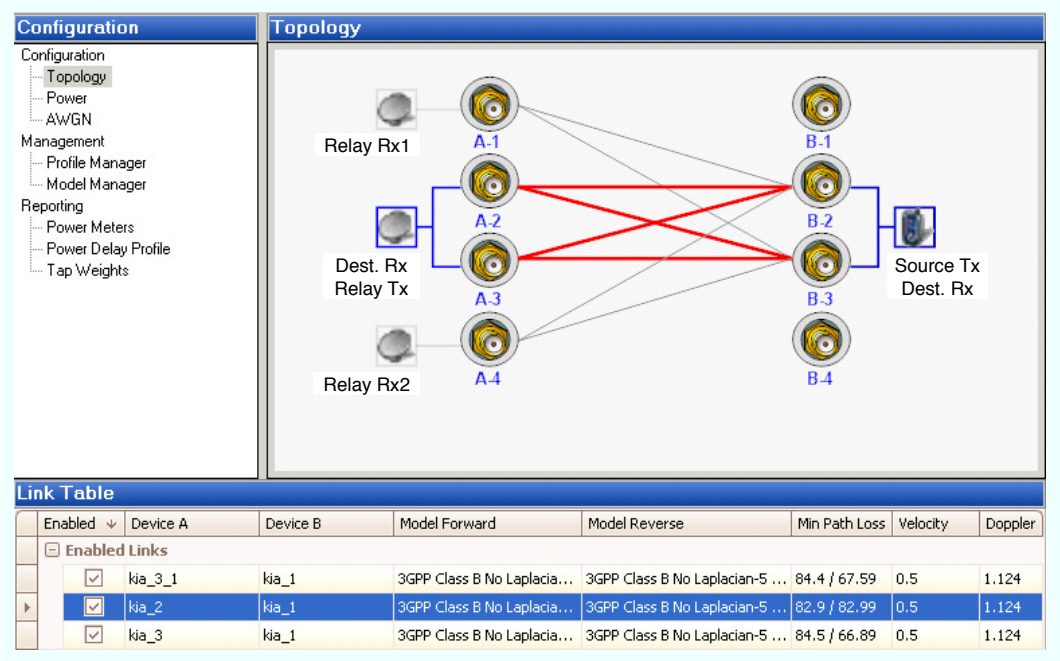

Figure 6.3 Node topology in the Azimuth Director software. Ports B2 and B3 correspond to the source transmitter in the first time slot. Also, A1 and A4 correspond to the relay two receive antennas in the first time slot, and A2 and A3 correspond to the destination receive antenna in the first time slot. In the second time slot, A2 and A3 are the relay transmit antennas and ports B2 and B3 serve as the destination receiver.

transmission mode. The path loss factors can, then, be computed as described in the earlier chapters using the following formulae:

$$\begin{aligned} \text{SNR}_{sr} &= \frac{\mu P}{(d_{sr})^{\alpha}}, \\ \text{SNR}_{rd} &= \frac{(1-\mu)P}{(d_{rd})^{\alpha}} \\ \text{SNR}_{sd} &= \frac{\mu P}{(d_{sd})^{\alpha}}, \end{aligned}$$

Figure 6.4 represents one particular representations, where the power losses on different paths correspond to $P_{loss,SR} = 17$ dB, $P_{loss,SD} = 33$ dB and $P_{loss,RD} = 28$ dB.

6.2 WARPLab Experiment Results

The hardware emulation results using the platform are shown in Figure 6.5 for a 2×2, 16-QAM system, where the relay is located at $d_{sr} = 0.2$, and the power splitting ratio is $\mu = 0.5$, and the channel is a 3GPP Class B channel [1]. Since the tests are performed on a hardware platform, the performance curves take into account the effects of the baseband processing as well as the RF chain, e.g., the amplifiers, the AGC (automatic gain control), imperfect channel estimate, etc. In the presence of such effects, the CPD method provides a middle point that improves the performance compared to the no-relay scenario, while avoiding the larger complexity of the FDF method, which conforms with the simulation results for other systems dimensions.

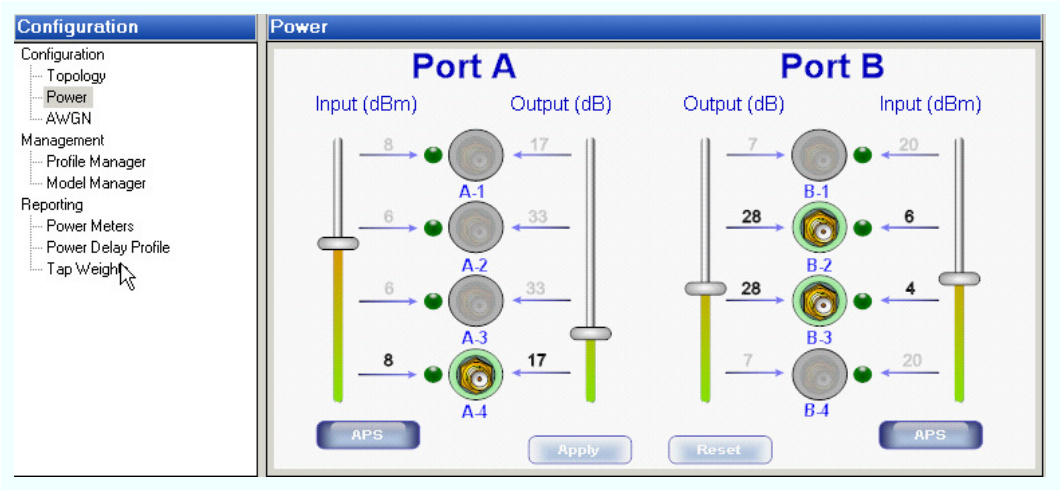

Figure 6.4 Setting the transmit and receive power based on the equivalent path loss.

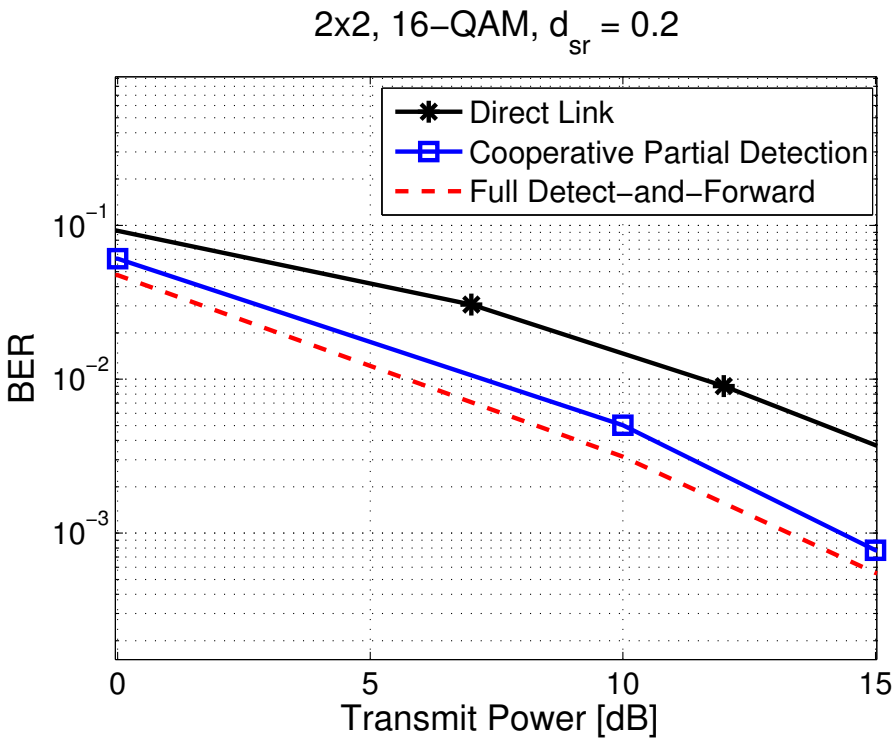

Figure 6.5 BER comparison of the no-relay, CPD and FDF techniques using the WARP hardware platform at the 2.4 GHz band. The channel emulation is done using the Azimuth ACE 400 WB [1] channel emulator, and the results include the RF effects.

Figure 6.6 shows the BER performance versus the expansion factor for different relay locations using a 3GPP Class B channel. The Class B channel represents a multi-tap pedestrian model. As the figure shows, the BER shows similar behavior for different relay locations. Moreover, Figure 6.7 shows how this performance changes when the Class A channel (single-tap channel) is used.

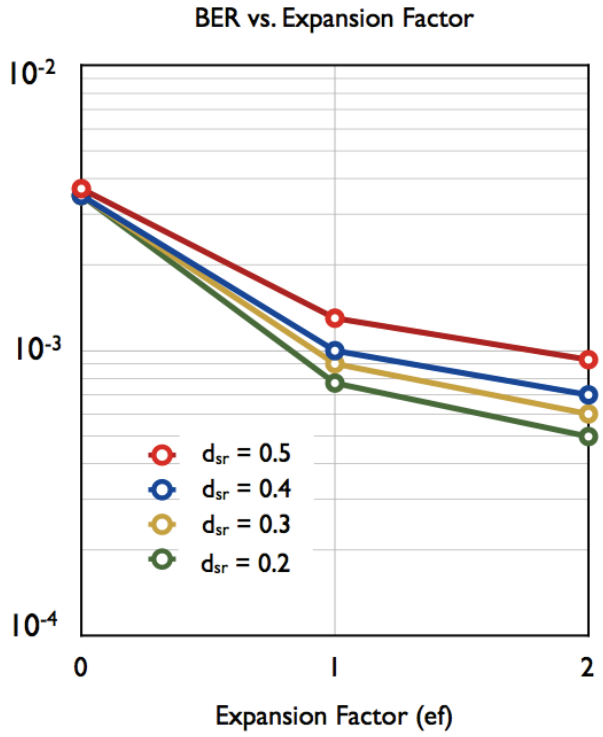

Figure 6.6 BER comparison of the no-relay, $ef = 0$, CPD, $ef = 1$, and FDF, $ef = 2$, techniques using the WARP hardware platform for different Expansion Factors using Class B channel model.

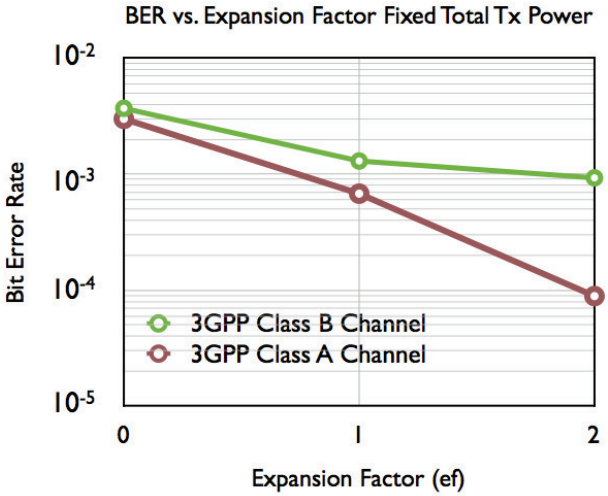

Figure 6.7 BER comparison of the no-relay, $ef = 0$, CPD, $ef = 1$, and FDF, $ef = 2$, techniques using the WARP hardware platform for both Class A and Class B channels when $d_{sr} = 0.5$.

Chapter 7
Conclusion and Future Work

7.1 Conclusion of the Current Results

In this thesis, we introduced reduced-complexity architectures for MIMO detectors based on linear and non-linear techniques, e.g., sphere detection and MMSE. Moreover, we proposed a novel configurable and flexible multi-user MIMO detector architecture, which can support different number of antennas and modulation orders required by a wide variety of different standards, and hence, can be applied in different nodes in a cooperative system. We presented the FPGA implementation results, and the simulation results suggest that the performance is considerably close to the optimum ML detector.

We proposed a novel and practical cooperative partial detection (CPD) scheme for MIMO relay networks; furthermore, our proposed scheme was based on architecture-friendly MIMO detection scenarios. CPD utilized the inherent structure of the tree-based sphere detectors, and modifies the tree traversal so that instead of visiting all the levels of the tree, only a subset of the levels, thus a subset of the transmitted streams, are visited. We developed a detection scheme based on an MRC combining of the received vectors. We showed that this scheme can be used to distribute the computational processing between the source and the destination, and more importantly, the relay can avoid the considerable overhead of MIMO detection while

helping the source-destination link to improve its performance. Finally, we demonstrated over-the-air experiments of the CPD algorithm using the WARP platform.

7.2 Future Work

The following problems can be further investigated as future work:

1. The impact of the AGC on the detection algorithm: We have observed that selecting an accurate value for the Automatic Gain Control (AGC) unit plays an important role in ensuring high performance. Therefore, designing detection algorithms that are resilient to AGC errors will be an interesting problem.

2. Joint partial detection and decoding: Performing the partial detection in the relay prevents the relay from generating enough log-likelihood ratios (LLRs) that could be combined with the conventional partial relay decoding schemes. Therefore, designing a join partial detection/decoding algorithm is another open problem to address.

Last, but not least, next generation wireless networks will need to rely more heavily on cooperation. Therefore, the general problem of designing architectures for cooperative systems while considering area, time and power tradeoffs will remain an exciting and challenging problem. The ultimate goal will be formulating this design in terms of complexity versus performance gain and finding bounds and solutions for the different points within the complexity-performance map. An implementation-oriented approach, with a focus on FPGA and ASIC solutions, will help formulating

the complexity in a meaningful and useful way; hence, leading to better insights into understanding the complexity-performance map in cooperative systems.

References

1. *Azimuth Systems : http://www.azimuthsystems.com/.*

2. *Coded Modulation Library : http://www.iterativesolutions.com/Matlab.htm.*

3. *IMT-Advanced Standard: http://www.ieee802.org/21/doctree/IMT-Advanced/18-07-00xx-00-0000_IMT_ Advanced_d3.doc.*

4. *WARP Testbed: https://www.warp.rice.edu/.*

5. *WINNER Channel Model: https://www.ist-winner.org/.*

6. *Xilinx FPGA Systems: https://www.xilinx.com/.*

7. A. Burg, *VLSI circuits for MIMO communication systems*, PhD Thesis (2006).

8. A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner and H. Bolcskei, *VLSI implementation of MIMO detection using the sphere decoding algorithm*, IEEE JSSC **40** (2005), no. 7, 1566–1577.

9. A. Sabharwal A. Chakrabarti and B. Aazhang, *Sensitivity of achievable rates for half-duplex relay channel*, IEEE Workshop on SPAWC (2005).

10. A. Sabharwal A. Chakrabarti, A. de Baynast and B. Aazhang, *LDPC code design for half-duplex decode-and-forward relaying*, Allerton Conference (2005).

11. A. Sabharwal B. Aazhang A. Chakrabarti, A. de Baynast, *Half-duplex estimate-and-forward relaying: bounds and code design*, ISIT (2006).

12. A. Chakrabarti, A. de Baynast, A. Sabharwal and B. Aazhang, *Low density parity check codes for the relay channel*, IEEE. J. Select. Areas Commun. **25** (2007), no. 2.

13. A. Sabharwal A. de Baynast, A. Chakrabarti and B. Aazhang, *A systematic construction of LDPC codes for relay channel in time-division mode*, Asilomar conference on signals, systems and computers (2006).

14. T.E. Hunter A. Nosratinia and A. Hedayat, *Cooperative communication in wireless networks*, IEEE Communications Magazine **42** (2004), no. 10, 68–73.

15. D. Goeckel A. Scaglione and J. N. Laneman, *Cooperative communications in mobile Ad-Hoc networks: rethinking the link abstraction*, IEEE Signal Processing Mag. **23** (2006), no. 5, 18–29.

16. A. Sendonaris, E. Erkip and B. Aazhang, *User cooperation diversity - part I: System Discription*, IEEE Trans. on Communications **51** (2003), no. 11, 1927–1938.

17. ———, *User cooperation diversity - part II: Implementation aspects and performance analysis*, IEEE Trans. on Communications **51** (2003), no. 11, 1939–1948.

18. J. N. Laneman A. Zaidi, S. Kotagiri and L. Vandendorpe, *Cooperative relaying with state available non-causally at the relay*, IEEE Trans. Inform. Theory **56** (2010), no. 5, 2272–2298.

19. K. Amiri, Yang Sun, P. Murphy, C. Hunter, J.R. Cavallaro, and A. Sabharwal, *WARP, a Unified Wireless Network Testbed for Education and Research*, IEEE International Conference on Microelectronic Systems Education, 2007., June 2007, pp. 53–54.

20. B. Farhang-Boroujeny, H. Zhu and Z. Shi, *Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems*, IEEE Transactions on Signal Processing **54** (2006).

21. B. Hochwald and S. ten Brink, *Achieving near-capacity on a multiple-antenna channel*, IEEE Transactions on Communications **51** (2003), 389–399.

22. B. Lu, G. Yue and X. Wang, *Performance analysis and design optimization of LDPC-coded MIMO OFDM systems*, IEEE Transactions on Signal Processing (2004).

23. B. Wang, J. Zhang and A. Host-Madsen, *On the capacity of MIMO relay channels*, IEEE Trans. on Info Theory **51** (2005), no. 1, 29–43.

24. C. K. Lo, S. Vishwanath and R. W. Heath, *Rate bounds for MIMO relay channels using precoding*, IEEE Global Telecommunications Conference (2005).

25. C. P. Schnorr and M. Euchner, *Lattice basis reduction: improved practical algorithms and solving subset sum problems*, Math. Programming **66** (1994), no. 2, 181–191.

26. A. M. Chan and I. Lee, *A new reduced-complexity sphere decoder for multiple antenna systems*, Proc. ICC (2002).

27. D. Chen and J. N. Laneman, *Modulation and demodulation for cooperative diversity in wireless systems*, IEEE Trans. Wireless Commun. **5** (2006), no. 7, 1785–1794.

28. D. N. C. Tse, *Fundamentals of Wireless communications*, Cambridge University Press, 2005.

29. P. K. Willett D. Pham, K. R. Pattipati and J. Luo, *An improved complex sphere decoder for V-BLAST systems*, IEEE Signal Processing Letters **11** (2004), no. 9, 748–751.

30. J. Rinas D. Wubben, R. Bohnke and V. Kuhn, *Efficient algorithm for decoding layered space-time codes*, IEE Electronics Letters **37** (2001), no. 22, 1348–1350.

31. L.M. Davis, *Scaled and decoupled Cholesky and QR decompositions with application to spherical MIMO detection*, IEEE Wireless Communications and Networking, vol. 1, 2003, pp. 326–331.

32. E. C. van der Meulen, *Transmission of information in a T-terminal discrete memoryless channel*, PhD Thesis, Univ. California, Berkeley, CA (1968).

33. ———, *Three-terminal communication channels*, Advanced Applied Probability **3** (1971), 120–154.

34. E. Viterbo and J. Boutros, *A universal lattice decoder for fading channels*, IEEE Trans. Inf. Theory **45** (1999), no. 5.

35. E. W. Jang, J. Lee, H. H. Lou and J. M Cioffi, *Optimal combining schemes for MIMO systems with hybrid ARQ*, International Symposium on Information Theory (2007), 2286–2290.

36. W. Rave E. Zimmermann and G. Fettweis, *On the complexity of sphere decoding*, International Symposium on Wireless Personal Multimedia Communications (2004).

37. C. Eklund, R.B. Marks, K.L. Stanwood, and S. Wang, *IEEE standard 802.16: a technical overview of the WirelessMAN air interface for broadband wireless access*, IEEE Communications Magazine **40** (2002), no. 6, 98–107.

38. H. Ekstrom, A. Furuskar, J. Karlsson, M. Meyer, S. Parkvall, J. Torsner, and M. Wahlqvist, *Technical solutions for the 3G long-term evolution*, IEEE Communications Magazine (2006).

39. Y. Fan and J. Thompson, *MIMO configurations for relay channels: Theory and practice*, IEE Trans. Wireless Commun. **6** (2007), no. 5, 1774–1786.

40. G. H. Golub and C. F. V. Loan, *Matrix computations*, 3rd ed., John Hopkins University Press, 1996.

41. G. J. Bradfod, J. N. Laneman, *A survey of implementation efforts and experimental design for cooperative communications*, Proceedings of ICASSP (2010).

42. G. J. Foschini and M. J. Gans, *On limits of wireless communications in a fading environment when using multiple antennas*, Wireless Personal Communications **6** (1998), 311–335.

43. J. Gotze and U. Schwiegelshohn, *A Square Root and Division Free Givens Rotation for Solving Least Squares Problems on Systolic Arrays*, SIAM Journal on Scientific and Statistical Computing **12** (1991), no. 4, 800–807.

44. H. Bolcskei, R. U. Nabar, O. Oyman and A. J. Paulraj, *Capacity scaling laws in MIMO relay networks*, IEEE Trans. on Wireless Comm. **5** (2006), no. 6, 1433–1444.

45. K. Higuchi H. Kawai and N. Maeda, *Likelihood function for QRM-MLD suitable for soft-decision turbo decoding and its performance for OFCDM MIMO multiplexing in multipath fading channel*, IEEE Trans Commun. (2005), no. 1, 47–56.

46. H. Long H. Zhao and W. Wang, *Reduced-Complexity Shortest Path-Sphere Decoder with Soft-Decision for Coded MIMO Systems*, ICCCAS (2006).

47. H. Zhu, B. Farhang-Boroujeny and R. R. Chen, *On performance of sphere decoding and Markov chain Monte Carlo detection methods*, IEEE Signal Processing Letters (2005).

48. H. Zhu, Z. Shi and B. Farhang-Boroujeny, *MIMO detection using Markov chain Monte Carlo techniques for near-capacity performance*, IEEE Internat. Conference on Acoustics, Speech and Signal Processing (2005).

49. A. Host-Madsen, *On the capacity of wireless relaying*, Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th, vol. 3, 2002, pp. 1333 – 1337 vol.3.

50. Y.H. Hu, *CORDIC-based VLSI architectures for digital signal processing*, Signal Processing Magazine, IEEE **9** (1992), no. 3, 16 –35.

51. T. Hunter and A. Nosratinia, *Diversity through coded cooperation*, IEEE Transactions on Wireless Communications **5** (2006), no. 2, 283–289.

52. I. E. Telatar, *Capacity of multiantenna Gaussian channels*, Europ. Trans. Telecommun. **10** (1999), 585–595.

53. J. Jie, C. Tsui and W. Mow, *A threshold-based algorithm and VLSI architecture of a K-best lattice decoder for MIMO systems*, ISCAS **4** (2005), 3359–3362.

54. J. Ma and X. Huang, *A system-on-programmable chip approach for MIMO sphere decoder*, IEEE Symposium on Field-Programmable Custom Computing Machines (2005), 317–318.

55. J. N. Laneman, *Cooperative diversity in wireless networks: algorithms and architectures*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

56. J. N. Laneman and G. W. Wornell, *Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks*, IEEE Trans. on Info Theory **49** (2003), no. 10, 2415–2425.

57. J. N. Laneman, D. N. C. Tse and G. W. Wornell, *Cooperative diversity in wireless networks: Efficient protocols and outage behavior*, IEEE Trans. on Info Theory **50** (2004), no. 12, 3062–3080.

58. J. Proakis, *Digital communications*, 4th ed., McGraw-Hill, 2001.

59. M. Jiang, S. X. Ng, and L. Hanzo, *Hybrid iterative multiuser detection for channel coded space division multiple access OFDM systems*, IEEE Transactions on Vehicular Technology **55** (2006), 115–127.

60. K. Amiri and J. R. Cavallaro, *FPGA implementation of dynamic threshold sphere detection for MIMO systems*, 40th Asilomar Conf on Signals, Systems and Computers (2006), 94–98.

61. _____, *Partial detection for multiple antenna cooperation*, Proceedings of CISS (2009).

62. _____, *Physical layer algorithm and hardware verification of MIMO Relays Using Cooperative Partial Detection*, Proceedings of ICASSP (2010).

63. K. Amiri, C. Dick, R. Rao and J. R. Cavallaro, *Novel sort-free detector with modified real-valued decomposition (M-RVD) ordering in MIMO systems*, Proc. of IEEE Globecom (2008).

64. _____, *A High Throughput Configurable SDR Detector for Multi-user MIMO Wireless Systems*, Springer Journal of Signal Processing Systems (2009).

65. K. Amiri, P. Radosaljevic and J. R. Cavallaro, *Architecture and algorithm for a stochastic soft-output MIMO detector*, 41th Asilomar Conf on Signals, Systems and Computers (2007), 1034–1038.

66. K. Wong, C. Tsui, R. S. Cheng and W. Mow, *A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels*, IEEE Int. Symp. Circuits Syst. **3** (2002), 273–276.

67. L. G. Barbero and J. S. Thompson, *A fixed-complexity MIMO detector based on the complex sphere decoder*, Signal Processing Advances in Wireless Communications, 2006. SPAWC '06. IEEE 7th Workshop on (2006).

68. _____, *FPGA design considerations in the implementation of a fixed-throughput sphere decoder for MIMO systems*, Field Programmable Logic and Applications, 2006. FPL '06. International Conference on (2006).

69. _____, *Performance analysis of a fixed-complexity sphere decoder in high-dimensional MIMO systems*, IEEE Conference on Acoustics, Speech and Signal Processing **4** (2006).

70. M. A. Khojastepour, *Distributed Cooperative Communications in Wireless Networks*, PhD Thesis, Rice University (2004).

71. M. A. Khojastepour, N. Ahmed and B. Aazhang, *Code design for the relay channel and factor graph decoding*, Proc. of Asilomar Conference **2** (2004), 2000–2004.

72. M. Janani, A. Hedayat, T. E. Hunter and A. Nosratinia, *Coded cooperation in*

wireless communications: space-time transmission and iterative decoding, IEEE. Trans. Signal Processing **52** (2004), 362–371.

73. M. Karkooti and J. R. Cavallaro, *Distributed decoding in cooperative communications*, Proc. of Asilomar Conference (2007), 824–828.

74. _____, *Cooperative communications using scalable medium block-length codes*, Proc. of WCNC Conference (2008).

75. M. Knox and E. Erkip, *Implementation of cooperative communications using software defined radios*, Proceedings of ICASSP (2010).

76. M. Myllyla., P. Silvola, M. Juntti and J. R. Cavallaro, *Comparison of two novel list sphere detector algorithms for MIMO-OFDM systems*, PIMRC (2006).

77. M. O. Damen, H. E. Gamal and G. Caire, *On maximum likelihood detection and the search for the closest lattice point*, IEEE Trans. on Inf. Theory **49** (2003), no. 10, 2389–2402.

78. R. Koetter M. Tuchler and A. C. Singer, *Turbo equalization: Principles and new results*, IEEE Transactions on Communications **50** (2002), 754–767.

79. M. Wenk, M. Zellweger, A. Burg, N. Felber and W. Fichtner, *K-best MIMO detection VLSI architecture achieving upto* 424 *Mbps*, IEEE ISCAS (2006), 1151–1154.

80. Y. Sun M. Wu and J. R. Cavallaro, *Reconfigurable real-time MIMO detector on GPU*, Asilomar Conference on Signals, Systems and Computers (2009).

81. ———, *Implementation of a 3GPP LTE turbo decoder accelerator on GPU*, IEEE Workshop on Signal Processing Systems (2010).

82. Y. Sun M. Wu, S. Gupta and J. R. Cavallaro, *A GPU implementation of A real-time MIMO detector*, IEEE Workshop on Signal Processing Systems (2009).

83. ———, *Implementation of a high throughput soft MIMO detector on GPU*, Journal of Signal Processing System (2010).

84. T. Meng, E. Lee, and D. Messerschmitt, *Least squares computation at arbitrarily high speeds*, Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87., vol. 12, apr 1987, pp. 1398 – 1401.

85. N. Moezzi-Madani and W. R. Davis, *Low-complexity high-throughput MIMO detector based on K-best algorithm*, ACM Great Lake Symposium on VLSI (2009), 451–456.

86. ———, *Parallel merge algorithm for high-throughput signal processing applications*, IET Electronics Letters **45** (2009), no. 3, 188–189.

87. ———, *Algorithm and Hardware Complexity Reduction Techniques for K-best Sphere Decoders*, ACM Great Lake Symposium on VLSI (2010).

88. P. Murphy, C. Hunter, and A. Sabharwal, *Design of a cooperative OFDM transceiver*, Proc. 2009 Asilomar Conference on Signals and Systems, 2009.

89. P. Murphy, A. Sabharwal, and B. Aazhang, *On building a cooperative communication system: testbed implementation and first results*, EURASIP Journal on Wireless Communications and Networking **2009** (2009).

90. T. Thorolfsson N. Moezzi-Madani and W. R. Davis, *A low-area flexible MIMO detector for WiMAX/WiFi standards*, IEEE conference on Design, Automation and Test in Europe (2010).

91. A. Nosratinia and T. E. Hunter, *Grouping and partner selection in cooperative wireless networks*, IEEE Journal on Selected Areas in Communications **25** (2007), no. 2, 369–378.

92. P. Radosavljevic and J. R. Cavallaro, *Soft sphere detection with bounded search for high-throughput MIMO receivers*, 40th Asilomar Conf on Signals, Systems and Computers (2006).

93. P. Silvola, K. Hooli and M. Juntti, *Suboptimal soft-output MAP detector with lattice reduction*, IEEE Signal Processing Letters **13** (2006), no. 6.

94. R. Vaze and R. W. Heath, *Capacity Scaling for MIMO Two-Way Relaying*, June 2007, pp. 1451–1455.

95. A. Sabharwal and U. Mitra, *Bounds and protocols for a rate-constrained relay channel*, Information Theory, IEEE Transactions on **53** (2007), no. 7, 2616 –2624.

96. Chitranjan K. Singh, Sushma H. Prasad, and Poras T. Balsara, *A Fixed-Point Implementation for QR Decomposition*, Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on, oct. 2006, pp. 75 –78.

97. Y. Sun and J. R. Cavallaro, *A New MIMO Detector Architecture Based on a Forward-Backward Trellis Algorithm*, IEEE Asilomar (2008), 1892 – 1896.

98. ———, *High Throughput VLSI Architecture for Soft-Output MIMO Detection Based on a Greedy Graph Algorithm*, ACM Great Lakes Symposium on VLSI Design (GLSVLSI'09) (2009).

99. ———, *Low-Complexity and High-Performance Soft MIMO Detection based on Distributed M-Algorithm Through Trellis-Diagram*, IEEE ICASSP (2010), 3398 – 3401.

100. T. E. Hunter, A. Nosratinia, *Cooperation diversity through coding*, IEEE Int. Symp. Info. Theory (2002).

101. R. Matsumoto T. Fukatani and T. Uyematsu, *Two methods for decreasing the*

computational complexity of the MIMO ML decoder, IEEE Trans Fundamentals (2004), no. 10, 2571–2576.

102. S. Sanayei T. Hunter and A. Nosratinia, *Outage analysis of coded cooperation*, IEEE Transactions on Information Theory **52** (2006), no. 2, 375–391.

103. T. M. Cover and A. A. El Gamal, *Capacity theorems for the relay channel*, IEEE. Trans. Info. Theory **IT-25** (1979), no. 5, 572–584.

104. T. Tang, C. B. Chae, R. W. Heath and S. Cho, *On achievable sum rates for a multiuser MIMO relay channel*, IEEE International Symposium on Info. Theory (2006).

105. X. Tang and Y. Hua, *Optimal Design of Non-Regenerative MIMO Wireless Relays*, IEEE Transactions on Wireless Communications **6** (April 2007), no. 4, 1398–1407.

106. R. Tannious and A. Nosratinia, *Spectrally efficient relay selection with limited feedback*, IEEE Journal on Selected Areas in Communication **26** (2008), no. 8, 1419–1428.

107. U. Fincke and M. Pohst, *Improved methods for calculating vectors of short length in a lattice, including a complexity analysis*, Math. Computat. **44** (1985), no. 170, 463–471.

108. S. Sun Y. Dai and Z. Lei, *Performance comparison between coded V-BLAST and GSTTC MIMO-OFDM systems*, 15th IEEE International Symp. on Personal Indoor and Mobile Radio Communications (2004), 2581–2585.

109. ———, *A comparative study of QRD-M detection and sphere decoding for MIMO-OFDM systems*, 16th IEEE International Symp. on Personal Indoor and Mobile Radio Communications (2005), 186–190.

110. Y. H. Nam, L. Liu, Y. Wang, C. Zhang, J. Cho and Jin-Kyu Han, *Cooperative communication technologies for LTE-advanced*, Proceedings of ICASSP (2010).

111. M. Yuksel and E. Erkip, *Broadcast strategies for the fading relay channel*, vol. 2, Oct. 2004, pp. 1060 – 1065 Vol. 2.

112. Z Guo and P. Nilsson, *A 53.3 Mb/s 4×4 16-QAM MIMO decoder in 0.35μm CMOS*, IEEE Int. Symp. Circuits Syst. **5** (2005), 4947–4950.

113. Z. Guo and P. Nilsson, *Algorithm and implementation of the K-Best sphere decoding for MIMO detection*, IEEE JSAC **24** (2006), no. 3, 491–503.